MEMORANDUM REPORT ARBRL-MR-03142

A GENERAL CURVILINEAR GRID GENERATION PROGRAM FOR PROJECTILE CONFIGURATIONS

J. L. Steger
C. J. Nietubicz
K. R. Heavey

October 1981

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22151.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MEMORANDUM REPORT ARBRL-MR-03142	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A GENERAL CURVILINEAR GRID GENERATION PROGRAM FOR PROJECTILE CONFIGURATIONS		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J.L. Steger, C.J. Nietubicz, & K.R. Heavey		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Ballistic Research Laboratory (ATTN: DRDAR-BLL) Aberdeen Proving Ground, Maryland 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L161102AH43
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Research & Development Command US Army Ballistic Research Laboratory (DRDAR-BL) Aberdeen Proving Ground, MD 21005		12. REPORT DATE OCTOBER 1981
		13. NUMBER OF PAGES 83
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Numerical grid generation	Finite difference techniques
Non-Orthogonal grids	Projectile shapes
Axisymmetric bodies	Elliptic equations
Transformed coordinates	

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A planar grid generation routine has been developed for use with standard and non-conventional projectile shapes. Three-dimensional grid generation has been obtained by generating a sequence of planar grids about axis normal cross-sections. The method is basically automatic and generates smoothly varying grids for arbitrary body shapes and allows for grid point clustering. The program is modular and can be used to generate planar Cartesian-like grids, C-grids, O-grids, or any portion thereof. The routine can be used for axisymmetric projectiles with or without stings, symmetric tubular projectiles

20. ABSTRACT (Continued)

and for any configuration with an axisymmetric nose. Sample grids are presented along with the program listing and sample output.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

Modern finite difference procedures for solving the partial differential equations which describe fluid flow frequently utilize curvilinear mapping procedures. Because boundary surfaces in the physical plane can be mapped onto rectangular surfaces in the transformed plane, a finite difference algorithm for the transformed equations can be readily applied to a variety of different body shapes. Even unsteady body motion is easily incorporated into the governing equations. To take advantage of the generality of the transformed equations, however, one needs a fairly automatic method of generating smoothly varying grids that fit arbitrary bodies and allow grid point clustering. The problem of grid generation, restricted to arbitrary projectile shapes, is the subject of this report.

A modular general purpose grid generation routine has been written for use with standard and nonconventional projectiles shapes. Three-dimensional grid generation capability is satisfied by generating a sequence of planar grids about axis normal cross sections. The grid generation routine can be used to generate planar Cartesian-like grids, C-grids, O-grids, or any portion thereof. The routine can be used for axisymmetric projectiles with or without stings, symmetric tubular projectiles, and for any projectile with an axisymmetric nose.

The modular grid generation program developed here contains its own body surface representation and grid point distribution routines. Another set of routines allows the user to build up an arbitrary outer boundary curve and grid point distribution. Finally the mesh itself is formed using either algebraic straight-line rays to connect inner and outer boundary points, or by using an elliptic solver. Clustering along grid lines is accomplished with an exponential clustering routine that allows the user to specify a given grid point spacing along the inner boundary curve. The modular structure of the grid generation program allows the user to substitute alternate clustering and grid generation routines of his own design.

Figures 1-8 show the various classification of grids which the code is designed to handle. These include axisymmetric projectiles with or without a sharp leading edge and with or without an afterbody sting, Figures 1-3. Isolated boattails or flares can also be meshed (Figure 2b). The code can also treat tubular projectiles (ring airfoils) that are axisymmetric, Figure 4. The isolated axisymmetric blunt body problem, Figure 5, is actually a special case of Figure 1.

In Section II the mechanics of the grid generation routine are described. Various grids are displayed to illustrate the ideas. Additional discussion of the computer grids is given in Section III, while complete documentation of the codes is given in the Appendices. The option of selecting a 2 or 3 dimensional grid is included and a plotting routine is presented in the appendix.

II. GRID GENERATION

The purpose of the grid generation routine is to generate a network of constant lines of ξ and η in the physical x-y plane as indicated in Figure 9a. Corresponding uniform values of ξ and η in the computational space define a one to one mapping between points j,k in the physical plane to points j,k in the computational plane, see Figure 9b. The mapping functions are described, at least numerically, once $\xi_{j,k}$ and $\eta_{j,k}$ are known in the physical plane as a function of $x_{j,k}$ and $y_{j,k}$; or conversely, once $x_{j,k}$ and $y_{j,k}$ are determined in the transform plane. The metric quantities ξ_x, ξ_y, η_x, and η_y needed in the transformed flow equations can then be determined numerically (see, for example, References 1-3).

In Figures 10a and 10b we show a typical grid generated for an axisymmetric projectile shape. In this case a spherical cap is placed at the end of the boattail (see Figure 10b) in order to avoid a highly discontinuous corner. The grid as it stands is suitable for axisymmetric or η-invariant[4] flow calculations. A three-dimensional grid can be formed by rotating the grid about the axis and defining constant lines in θ. When viewed in this manner the grid is equivalent to warping a spherical coordinate about a nonspherical projectile shape. Using the notation defined in Figure 9a, the axis corresponds to the $\xi = 0$ and $\xi = \xi_{max}$ lines. The inner and outer boundaries correspond to $\eta = 0$ and $\eta = \eta_{max}$.

Depending on the projectile shape a warped hemispherical coordinate may be used (Figure 1) or a warped Cartesian coordinate (Figure 3). A ring airfoil (tubular projectile) can be meshed with a C-grid (Figure 4). One could also spin an airfoil O-grid (Figure 7) about the tubular projectiles axis of symmetry. Whichever class of grid is used, however, it must map onto the uniform computational plane shown in Figure 9b. The $\eta = 0$ plane is not necessarily restricted to only the body surface. It may for example, include the forward cut of Figure 2a or the lower and upper cuts of Figure 4.

[1] *Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries", AIAA Journal, Vol. 16, July 1978, pp. 679-686.*

[2] *Pulliam, T. H. and Steger, J. L. "On Implicit Finite-Difference Simulations of Three-Dimensional Flow", AIAA Paper No. 78-10, 1978.*

[3] *Schiff, L. B. and Steger, J. L., "Numerical Simulation of Steady Super sonic Viscous Flow", AIAA Paper No. 79-130, 1979.*

[4] *Nietubicz, C. J., Pulliam, T. H., and Steger, J. L., "Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations", AIAA Paper No. 79-0010, 1979.*

In generating a projectile grid such as those indicated in Figures 1-8, one first decides what class of grid fits the given problem. The grid generation problem can then be broken into three main tasks as follows: (1) define the body shape, possible sting or cut, and distribute grid points along the $\eta = 0$ boundary (i.e., specify ξ as a function of x and z along $\eta = 0$). Points along this boundary should be clustered to flow field gradients, e.g., the forward stagnation point, expansions, shocks; (2) define the outer boundary curve and distribute grid points along the $\eta = \eta_{max}$ boundary. Here we restrict $\xi = 0$ and $\xi = \xi_{max}$ to be vertical or horizontal straightline rays in order to simplify programming logic, thus the endpoints of the η_{max} curve must properly align with those of the $\eta = 0$ boundary; (3) once the outer boundaries are defined, they are "connected" by generating the interior grid with appropriate clustering functions in η.

In the remainder of this Section the procedures used to generate the $\eta = 0$ boundary, the $\eta = \eta_{max}$ boundary, and the interior clustered grid, will be described.

A. Surface Representation and Grid Point Distribution

The first step in generating the grid is to represent and distribute points along the body surface. A sting or cut may also have to be included.

The body shape is expected to have either an analytic description or be described as a table of x,y ordinates. In either case the data is assumed to be nondimensional with respect to a reference length which can, of course, be taken as 1 so the data remains unaltered.

The present code allows for either a parabolic arc or standard class projectile, such as a projectile with a sharp tangent-ogive or blunt secant ogive nose, cylindrical body, boattail, and spherical cap. If the analytic body shape differs from the above mentioned shapes, then the user must supply his own description. In this case values of x along the body axis (or chord) will be distributed by contiguously combining segments of the clustering function

$$x_j = x_o + a\psi_j + b\psi_j^2 + c\psi_j^3 \qquad \begin{array}{c} x_o \leqslant x_j \leqslant x_f \\ j_o \leqslant j \leqslant j_f \end{array} \tag{1}$$

where $\psi_j = (j-j_o)/(j_f-j_o)$ and j is an index value such that points j_o to j_f lie in the interval x_o to x_f and $x_{j_o} = x_o$ while $x_{j_f} = x_f$. Equation (1) is used to cluster x_j as a function of j as indicated in Figure 11. The user determines the shape of the clustering function by specifying the initial and final increments of x, that is

$$\Delta x_o = x_{j_o+1} - x_{j_o} \tag{2a}$$

$$\nabla x_f = x_{j_f} - x_{j_f-1} \tag{2b}$$

Since x_o and x_f are also specified, a, b, and c are determined

$$c = \{\nabla x_f + \Delta x_o - 2h(x_f-x_o)\}/(h - 3h^2 + 2h^3)$$

$$b = \{\Delta x_o - h(x_f-x_o) - c(h^3 - h)\}/(h^2 - h)$$

$$a = x_f - x_o - b - c$$

where $h = (j_f - j_o)^{-1}$.

The amount of clustering at each point is determine by the specified values of Δx_o and ∇x_f. Moreover, because Δx_o and ∇x_f are specified, the user can smoothly patch functions together to form a general clustering function. Examples of this are indicated in the computer output presented in Appendix C. One drawback to the clustering function, Eq. (1), is that the function is not guaranteed to be monotone in the interval. This can happen, for example, if Δx_o is too small and ∇x_f too large. Again, the output in Appendix C indicates practical values to choose.

In the case of a nonanalytic body shape x and y are read in as a table of values. Here either an axis length or surface arc length is used as a clustering function. If the axis length is chosen, x is given and corresponding y values are found from the table of x,y coordinates using cubic spline interpolation[5]. Alternately, the surface arc length can be used. In this approach the arc length s is computed from the table of x,y values, and the length is normalized. A new normalized clustered arc length is then defined, using Eq. (1) with s (the arc length) in place of x. Both x and y are then interpolated from the tables x versus s and y versus s. Again cubic spline interpolation is used.

Finally, a sting or cut may be added to the configuration. Again, the clustering relation, Eq. (1), is used to distribute points along the sting or cut.

[5] *Dahlquist, G. and Bjorck, A., "Numerical Methods". Prentice-Hall, Englewood Cliffs, New Jersey, 1974.*

B. Outer Boundary Formation and Grid Point Distribution

Usually the shape of the outer boundary curve is not as well defined as the inner boundary. However, the shape of the outer curve may be partially prescribed. For example, in Figure 4, the axis of symmetry of the ring airfoil has a specified location. The side and upper boundaries need only be smooth curves far-removed from the body. If, however, the ring airfoil or standard projectile is tested in a wind tunnel, a numerical simulation of the experiment requires a fixed wall outer boundary. In this case the top portion of the outer boundary curve must be specified.

A part of the grid generation problem then is the formation of the arbitrary outer boundary. Here this boundary is built up by connecting contiguous cubic segments, which in the degenerate case can be straight lines. Figures 12a and 12b illustrate two typical outer boundary curves. In Figure 12a three cubic segments make up the boundary $\eta = \eta_{max}$. Each segment, from a to b for example, is formed by specifying the endpoints x,y, and angle θ, where θ is the angle between the curve and the x axis. In the example, Figure 12a, $\theta_a = 90°$, $\theta_b = \theta_c = 0°$ or 180° and $\theta_d = 90°$.

The data x,y,θ at each endpoint determines the shape of the parametric curves

$$\begin{aligned} x &= x_o + \alpha_1 t + \alpha_2 t^2 \\ y &= y_o + \beta_1 t + \beta_2 t^2 \end{aligned} \qquad 0 \leq t \leq 1 \tag{4}$$

which are equivalent to a cubic

$$y = y_o + \gamma_1(x-x_o) + \gamma_2(x-x_o)^2 + \gamma_3(x-x_o)^3 \tag{5}$$

The parametric cubic is used because the condition $\frac{dy}{dx} = \infty$ can be specified, e.g., segment bc of Figure 12b has this constraint at both endpoints.

The solution of the parameters α_1, α_2, β_1, and β_2 are given by

$$\begin{aligned} \alpha_1 &= m_o \\ \alpha_2 &= (x_f - x_o) - \alpha_1 \\ \beta_1 &= n_o \\ \beta_2 &= (y_f - y_o) - \beta_1 \end{aligned} \tag{6}$$

where

$$m_o = \left.\frac{dx}{dt}\right|_o \qquad m_1 = \left.\frac{dx}{dt}\right|_1$$

$$n_o = \left.\frac{dy}{dt}\right|_o \qquad n_1 = \left.\frac{dy}{dt}\right|_1$$

and

$$n_o = m_o \left.\frac{dy}{dx}\right|_o \quad \text{or} \quad m_o = n_o \left.\frac{dx}{dy}\right|_o$$

$$n_1 = m_1 \left.\frac{dy}{dx}\right|_1 \quad \text{or} \quad m_1 = n_1 \left.\frac{dx}{dy}\right|_1$$

The solutions for n_o, m_o, n_1, and m_1 are conditional insofar that infinite-slopes are avoided. Then the regular solutions are

i) If $y_x|_o < x_y|_o$, $y_x|_1 < x_y|_1$, and $y_x|_1 \neq y_x|_o$

$$\begin{aligned} m_1 &= 2\,[(x_f - y_o) - (x_f - x_o)\, y_x|_o]/(y_x|_1 - y_x|_o) \\ m_o &= 2\,(x_f - x_o) - m_1 \\ n_o &= m_o\, y_x|_o \\ n_1 &= m_1\, y_x|_1 \end{aligned} \qquad (7a)$$

ii) If $y_x|_o < x_y|_o$, $x_y|_1 < y_x|_1$, and $x_y|_1 y_x|_o \neq 1$

$$\begin{aligned} m_o &= 2[(x_f - x_o) - (y_f - y_o)\, x_y|_1]/(1 - x_y|_1\, y_x|_o) \\ n_1 &= 2\,(y_f - y_o) - m_o\, y_x|_o \\ n_o &= m_o\, y_x|_o \\ m_1 &= n_1\, x_y|_1 \end{aligned} \qquad (7b)$$

iii) If $x_y|_o < y_x|_o$, $y_x|_1 < x_y|_1$, and $y_x|_1 x_y|_o \neq 1$

$$\begin{aligned} n_o &= 2[(y_f - y_o) - (x_f - x_o)\, y_x|_1]/(1 - y_x|_1\, x_y|_o) \\ m_1 &= 2\,(x_f - x_o) - n_o\, x_y|_o \\ m_o &= n_o\, x_y|_o \\ n_1 &= m_1\, y_x|_1 \end{aligned} \qquad (7c)$$

iiii) If $x_y|_o < y_x|_o$, $x_y|_1 < y_x|_1$, and $x_y|_1 \neq x_y|_o$

$$
\begin{aligned}
n_1 &= 2\,[(x_f - x_o) - (y_f - y_o)\, x_y|_o]/(x_y|_1 - x_y|_o) \\
n_o &= 2\,(y_f - y_o) - n_1 \\
m_o &= n_o\, x_y|_o \\
m_1 &= n_1\, x_y|_1
\end{aligned}
\qquad (7d)
$$

Whenever the third constraint is violated (for example in case (i), if $y_x|_1 = y_x|_o$) a linear curve is used. In this case $\alpha_2 = \beta_2 = 0$ and $\alpha_1 = x_f - x_o$, $\beta_1 = y_f - y_o$. The segments ab, bc, and ea of Figure 12b are examples of the straight line segments.

The outer boundary curve is made up of contiguous cubic segments starting from the $\xi = 0$ boundary. Points are distributed along this curve either as a uniform distribution of arc length, or as a specified arc length distribution using the previously defined clustering scheme, Eq. (1). Since the true arc length is not specified a priori, precise alignment of points along the outer boundary can be specified only after the cubic segments are specified and the arc length is computed. Without such knowledge a normalized clustering function should be used.

C. Grid Generation and Clustering

The task of generating a grid is undertaken once the boundary curves are specified and points are distributed on the $\eta = 0$ and η_{max} boundaries. Two types of grid generation procedures were used and are discussed below.

In the first case, lines of constant ξ (i.e., the rays emerging from the body) are formed by simply connecting straight lines from points along $\eta = 0$ to points along $\eta = \eta_{max}$. The spacing in η along each such line is either uniform or is determined by the relation

$$\Delta s_k = \Delta s_o (1 + \varepsilon)^{k-1} \quad , \quad k = 1, k_{max} - 1 \qquad (8)$$

Here Δs_o is the specified constant grid spacing at the inner boundary. The parameter ε is determined by a Newton-Raphson iteration process so that the sum of the above increments matches the known arc length between the $\eta = 0$ and $\eta = \eta_{max}$ for points which have the same values of ξ. Figures 13a and 13b illustrate a straight ray grid with clustering in η for a tubular projectile.

In the second case, the grid is generated with elliptic partial differential equations following References 6, 7, 8. The grid generating equations are solved on the specified computational space for unknowns $x_{j,k}$ and $y_{j,k}$:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2 \; (\overline{P} x_{\xi} + \overline{Q} x_{\eta}) \tag{9a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2 \; (\overline{P} y_{\xi} + \overline{Q} y_{\eta}) \tag{9b}$$

where

$$\alpha = x_{\eta}^{2} + y_{\eta}^{2}, \quad \beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}, \quad \gamma = x_{\xi}^{2} + y_{\xi}^{2}, \quad J = x_{\xi}y_{\eta} - x_{\eta}y_{\xi}$$

and

$$\overline{P} = P_o \; e^{-a(\eta-\eta_o)} + P_m \; e^{-a(\eta-\eta_{max})} \quad .$$

$$\overline{Q} = Q_o \; e^{-b(\eta-\eta_o)} + Q_m \; e^{-b(\eta-\eta_{max})}$$

Here P_o, Q_o, P_m, Q_m, a and b are prescribed clustering parameters. Along the $\eta = 0$ and $\eta = \eta_{max}$ boundaries, $x_{j,k}$ and $y_{j,k}$ have been previously prescribed. Along the $\xi = 0$ and $\xi = \xi_{max}$, which are either vertical or horizontal lines in the physical space, the following boundary conditions are enforced: either

$$x \text{ is given and } y_{\xi} = 0 \tag{10a}$$

on a vertical boundary, or

$$x_{\xi} = 0 \text{ and } y \text{ is given} \tag{10b}$$

on a horizontal boundary.

[6] *Chu, W. H., "Development of a General Finite Difference Approximation for a General Domain". Journal of Comp. Physics, Vol. 8, 1971, pp. 392-408.*

[7] *Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies". Journal of Comp. Physics, Vol. 15, 1974, pp. 299-319.*

[8] *Sorenson, R. L. and Steger, J. L., "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations". NASA TM 73252, August 1977.*

For periodic grids as indicated in Figure 7, these boundary conditions in ξ are replaced by the usual periodic relations.

The derivative expressions on the left hand side of Eq. (9) are all differenced with conventional second order central difference operators, that is

$$
\begin{aligned}
x_\xi &= (x_{j+1,k} - x_{j-1,k})/(2\Delta\xi) \\
x_\eta &= (x_{j,k+1} - x_{j,k-1})/(2\Delta\eta) \\
x_{\xi\xi} &= (x_{j+1,k} - 2x_{j,k} + x_{j-1,k})/(\Delta\xi)^2 \\
x_{\xi\eta} &= (x_{j+1,k+1} - x_{j+1,k-1} - x_{j-1,k+1} + x_{j-1,k-1})/(4\Delta\xi\Delta\eta) \\
x_{\eta\eta} &= (x_{j,k+1} - 2x_{j,k} + x_{j,k-1})/(\Delta\eta)^2
\end{aligned}
\tag{11}
$$

while derivatives of y are treated identically. The Jacobian J is formed with central differencing. The right hand side companion terms to $\overline{P}$ and $\overline{Q}$, however, are backward or forward differenced depending on the sign of $\overline{P}$ and $\overline{Q}$. If $\overline{P}$ is positive, x_ξ and y_ξ are forward differenced. The terms x_η, y_η are differenced in the same way.

The one sided differencing for the right side term was chosen assuming J is a constant. Preliminary analysis with local linearization of terms like J^2x_ξ suggests one sided differencing should also be used in J to keep balanced coefficients. This however has not been evaluated.

The difference equations to Eq. (9) are solved with a successive line overrelaxation (SLOR) procedure. As an initial guess for the relaxation procedure we use the straight line ray procedure previously described. For the most part, if coefficients $\overline{P}$ and $\overline{Q}$ are large, the SLOR procedure is very difficult to converge. Consequently, we recommend using the algebraic clustering function, Eq. (8).

In the algebraic clustering approach the elliptic solver is used to generate a grid with $\overline{P} = \overline{Q} = 0$. The x,y points along a ξ = constant line are then redistributed along this line as a function of arc length. The clustering function Eq. (8) is used for this purpose. This procedure works quite well and provides excellent control of the grid spacing near the body surface. Further details are given in Reference 8. The grid shown in Figure 14 was generated in this manner.

The elliptic solver need not be used over the entire range in ξ. Because of the boundary condition, Eq. (10), the elliptic equations can be joined to a straight ray along any vertical or horizontal boundary line in ξ. Figure 15 shows details of such a procedure used in the previous tubular projectile case. Here the ξ-region over the tubular projectile is meshed using the elliptic equations while the remainder is meshed with straight rays. After the basic grid is formed, the entire grid is clustered in η using Eq. (8).

D. Grid Plotting

An integral part of the grid development program is the ability to plot the computed grid in a timely manner. The plot program which was developed and utilized allowed almost instantaneous viewing of the computed grid. This capability significantly reduced the grid generator development time.

The plot program was written using Tektronix Plot 10 software on the BRL Cyber 176 computer. A program listing is presented in Appendix D.

The only input required for the plotting program is the converged grid file and the minimum and maximum x,y values of the grid. The interactive program uses prompts for the remaining input.

III. DISCUSSION OF RESULTS AND CONCLUDING REMARKS

Figures 10, 13, 14, and 15 give the reader a reasonably clear picture of the capability of the grid generation routine. The other grids classified in Figures 1-8 simply use the same program elements in different arrangements.

An analytic shape that was meshed using the elliptic equation approach is illustrated in Figure 10. Along the body, points are clustered to the nose, boattail junctures, and base. No cuts or stings are used. Only two cubic segments are used to define the outer boundary and along this curve points are uniformly distributed. Solution of Eq. (9) with no additional reclustering completes the grid generation problem. Note that Eq. (10b) is well satisfied along the $\xi = 0$ and $\xi = \xi_{max}$ axis.

In Figure 13 a grid for a tubular projectile is shown. The body is defined by x,y ordinates and here upper and lower cuts are used. The outer boundary is defined using four cubic segments, two of which degenerate to straight lines. From the trailing edge on back, the point distribution along the outer boundary matches that of the cuts. In this way vertical rays are used over the cut, although this is not required. Straight line rays make up the interior grid, and along these rays points in η are exponentially clustered using Eq. (8). The controlled grid spacing along the body is illustrated in Figure 13b.

The case shown in Figure 14 is similar to Figure 10 only here the grid generated using Eq. (9) was reclustered along lines of constant ξ. The grid spacing near the body is now controlled as before.

Finally, the case shown in Figure 15 is similar to that of Figure 13 only now an elliptic solver is used over the airfoil. The cut region is again treated with vertical rays.

Grids for nonaxisymmetric bodies with axisymmetric noses, Figure 6, can be generated as follows. For the axisymmetric nose, Figure 5, the grid is generated in a plane and then spun around the axis forming a three-dimensional grid. The remaining grid can then be generated by taking planar cuts normal to the axis at various increments Δx (x aligned with the axis, see Figure

6). At each cut a planar grid is generated and the combinations of these grids form the three-dimensional mesh. At each cross section one generates the O-type grids shown in Figures 7 and 8 being careful to maintain continuity in x.

Completed computer code documentation is provided in Appendices A and B. Input and output to obtain the grid shown in Figure 15 is included in Appendix C and the plotting code is given in Appendix D.

The modular program developed here has proven to be quite flexible, and should find application in determining grids for various conventional and nonconventional projectile shapes.

REFERENCES

1. Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries", AIAA Journal, Vol. 16, July 1978, pp. 679-686.

2. Pulliam, T. H., and Steger, J. L., "On Implicite Finite-Difference Simulations of Three-Dimensional Flow", AIAA Paper No. 78-10, 1978.

3. Schiff, L. B., and Steger, J. L., "Numerical Simulation of Steady Supersonic Viscous Flow", AIAA Paper No. 79-130, 1979.

4. Nietubicz, C. J., Pulliam, T. H., and Steger, J. L., "Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations", AIAA Paper No. 79-0010, 1979.

5. Dahlquist, G. and Bjorck, A., "Numerical Methods". Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

6. Chu, W. H., "Development of a General Finite Difference Approximation for a General Domain". Journal of Comp. Physics, Vol. 8, 1971, pp. 392-408.

7. Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies". Journal of Comp. Physics, Vol. 15, 1974, pp. 299-319.

8. Sorenson, R. L., and Steger, J. L., "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations". NASA TM 73252, August 1977.

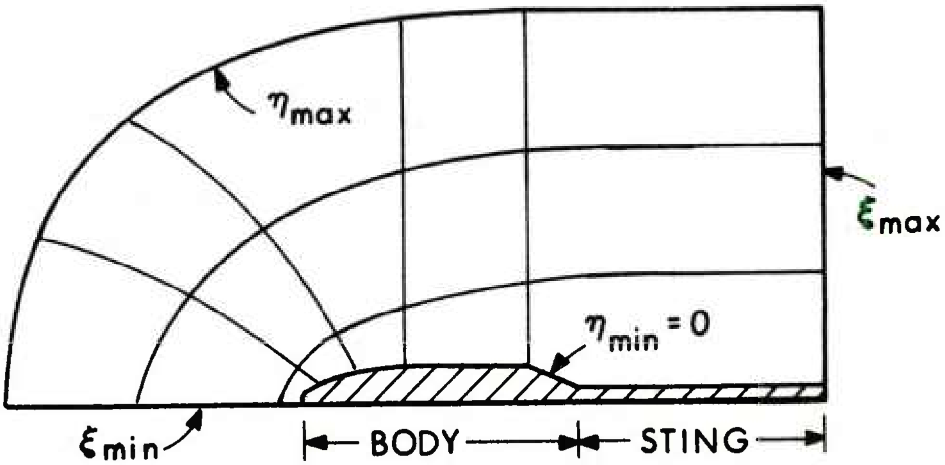


Figure 1. Standard Projectile Grid with Sting

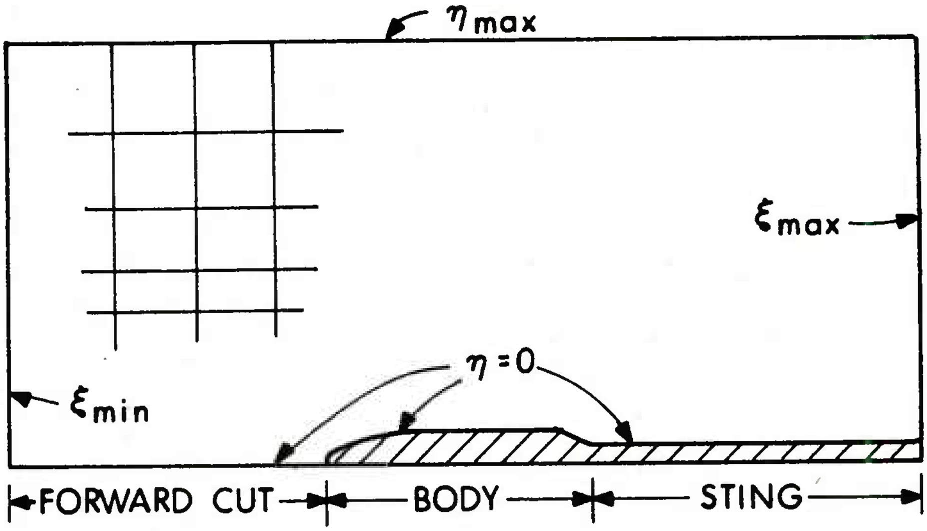


Figure 2a. Cartesian-like Projectile Grid with Sting

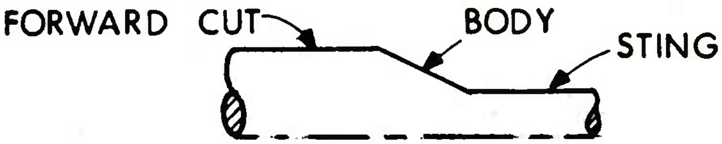


Figure 2b. Special Case Isolated Boattail

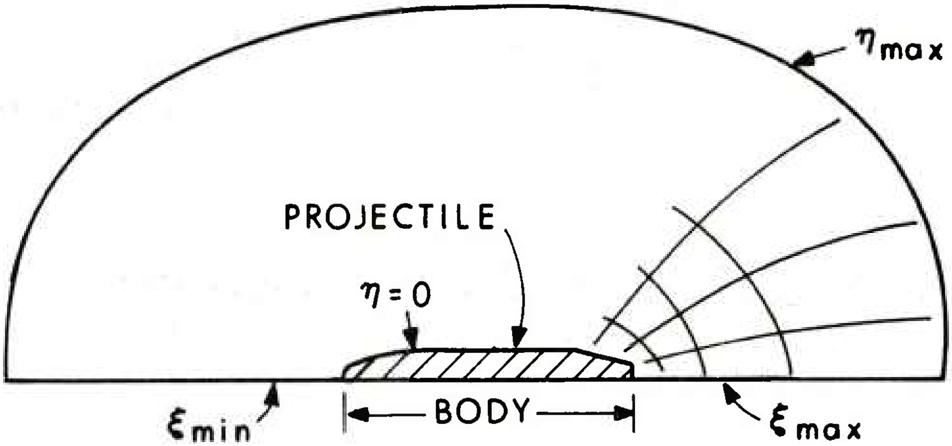


Figure 3. Standard Projectile Grid with Base, Special Case of O-Grid with Symmetry

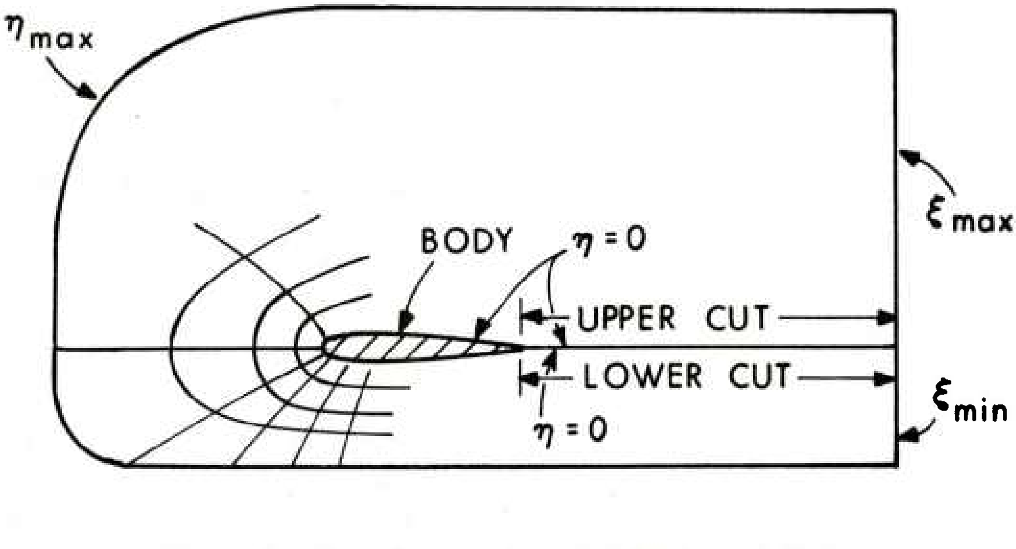


Figure 4. Tubular Projectile Grid or C-Grid

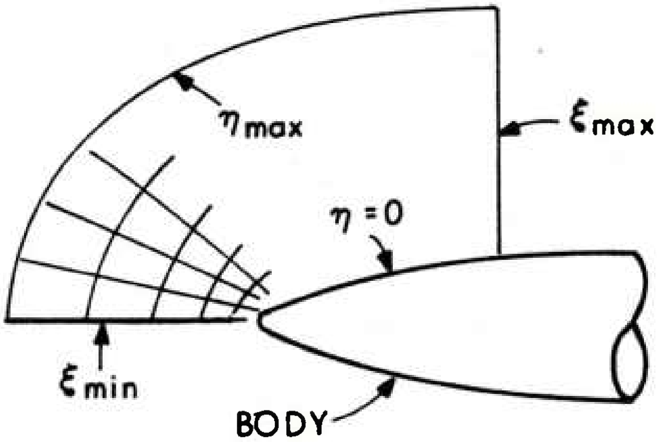


Figure 5. Projectile Blunt Body Grid (Fraction of O-Grid)

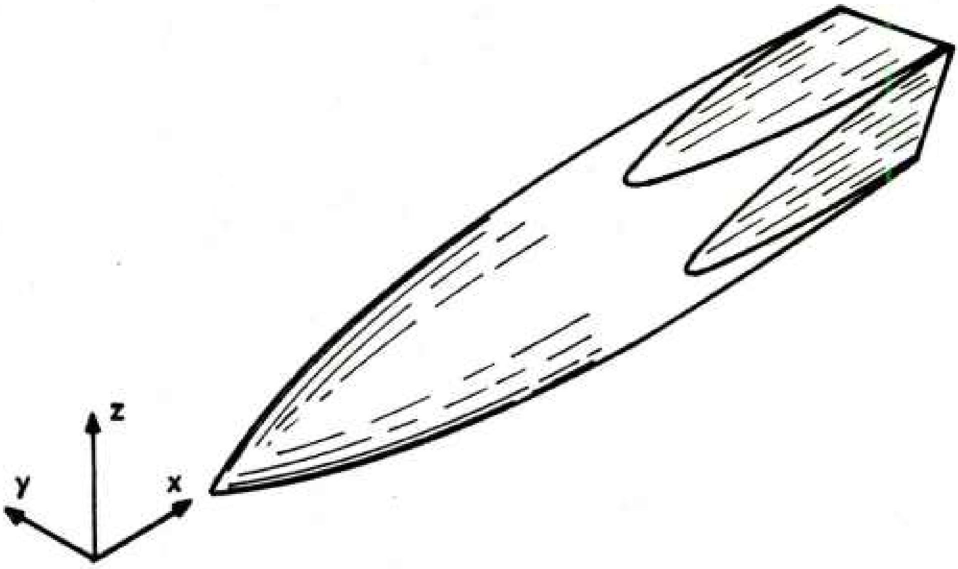


Figure 6. Projectile with Symmetric Nose and Nonsymmetric Afterbody

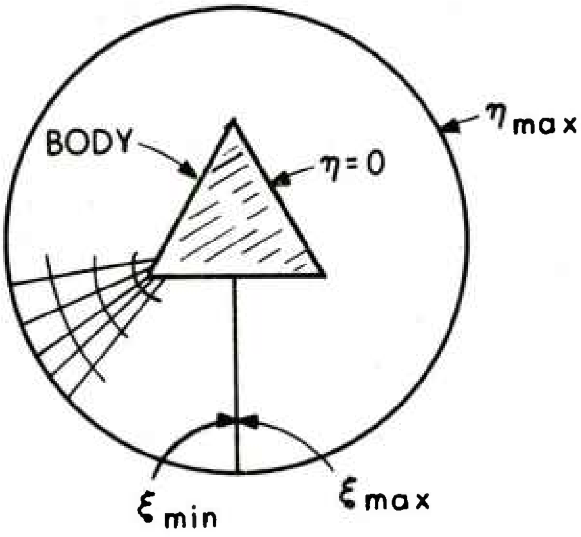


Figure 7. Projectile Cross Section with Periodic B.C. (O-Grid)

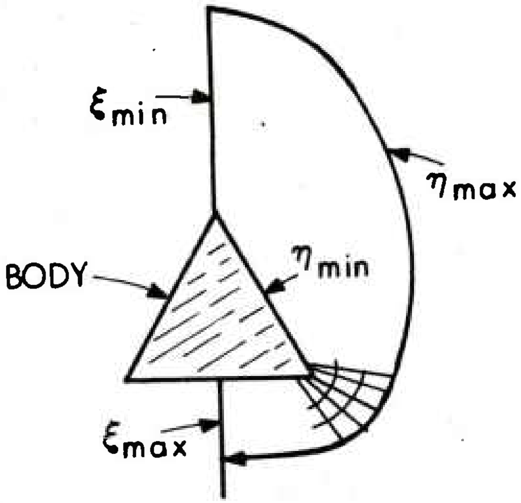


Figure 8. Projectile Cross Section with Symmetry Plane (O-Grid)

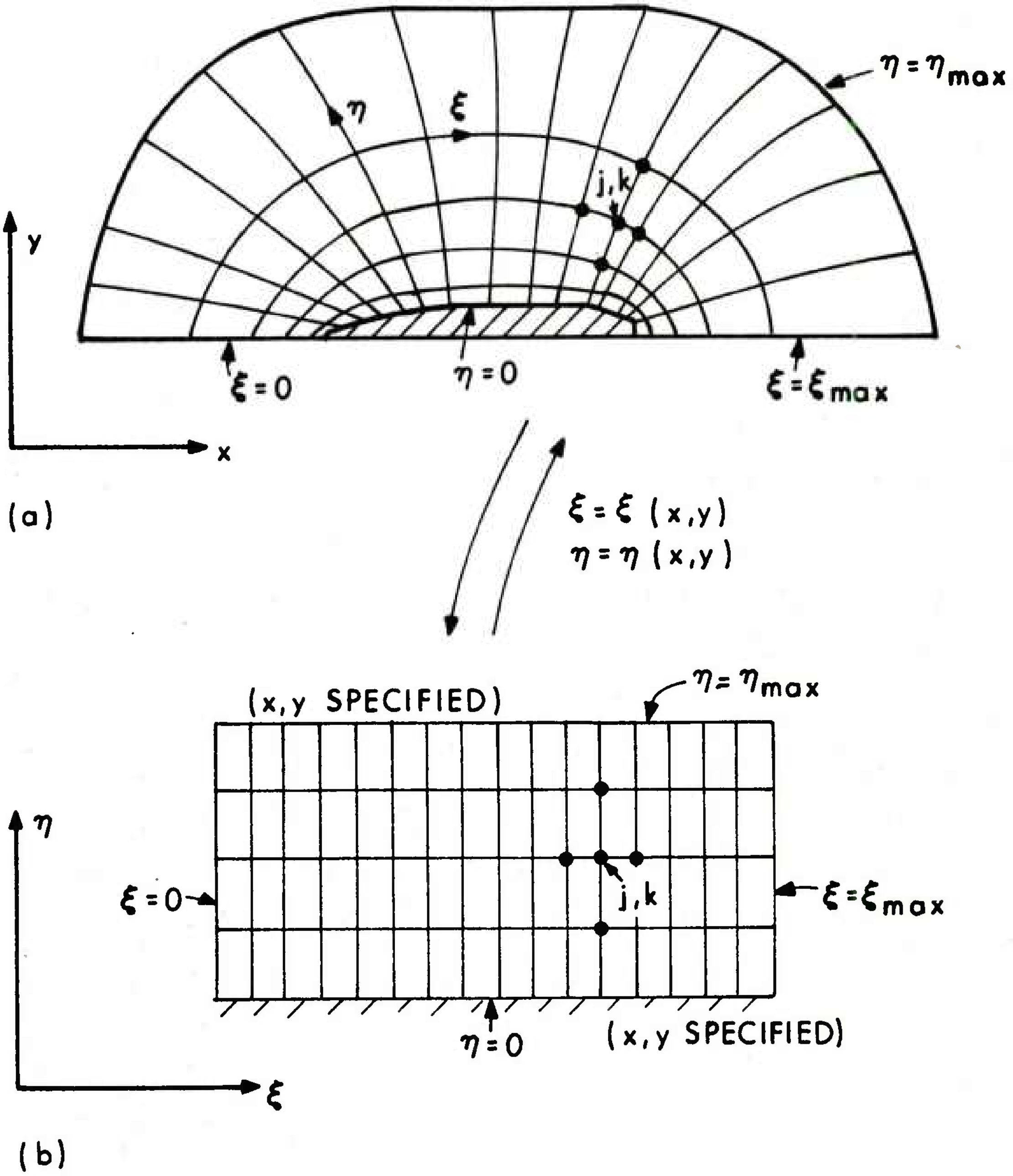


Figure 9. Mapping from Physical Space to Computational Space

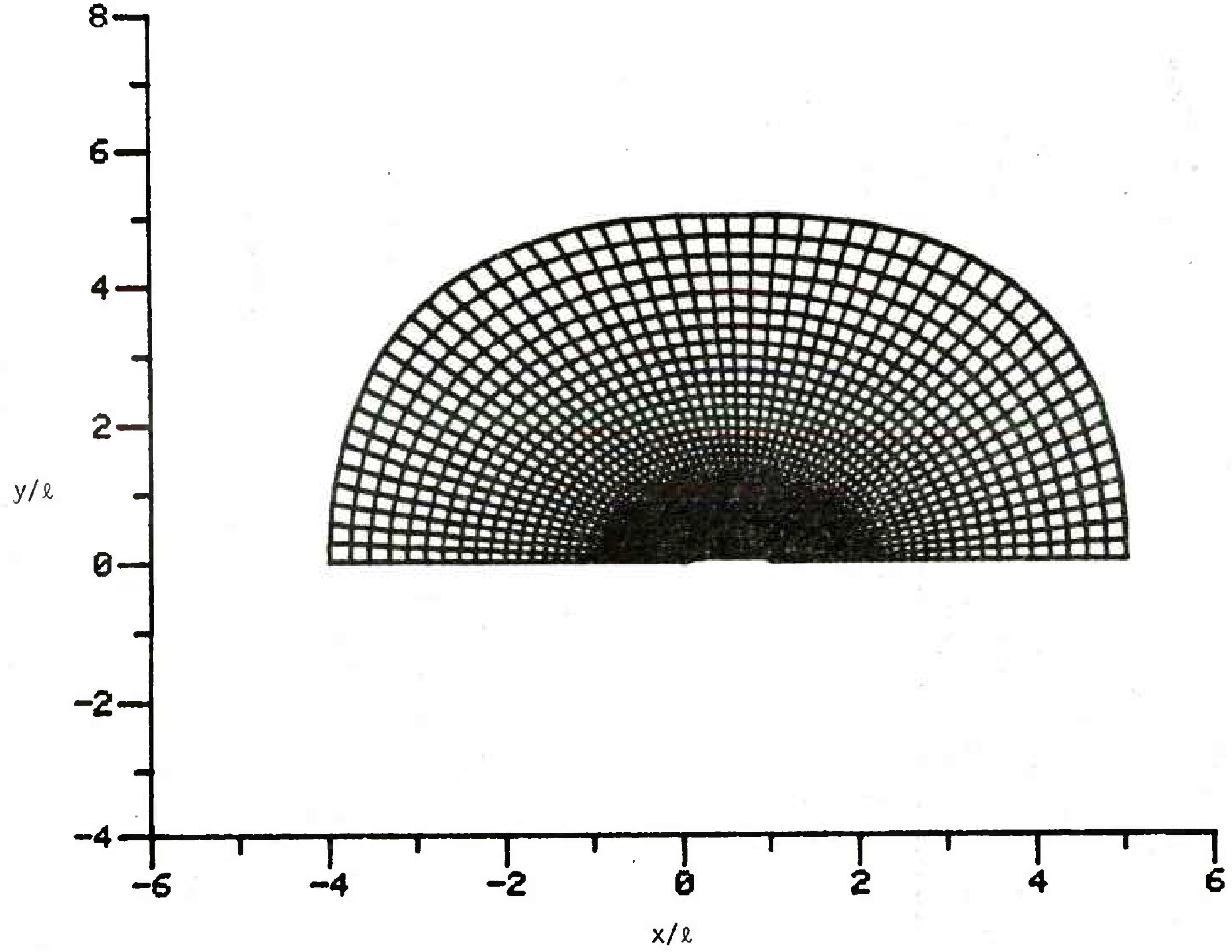


Figure 10a. Standard Projectile Grid in the Physical Plane

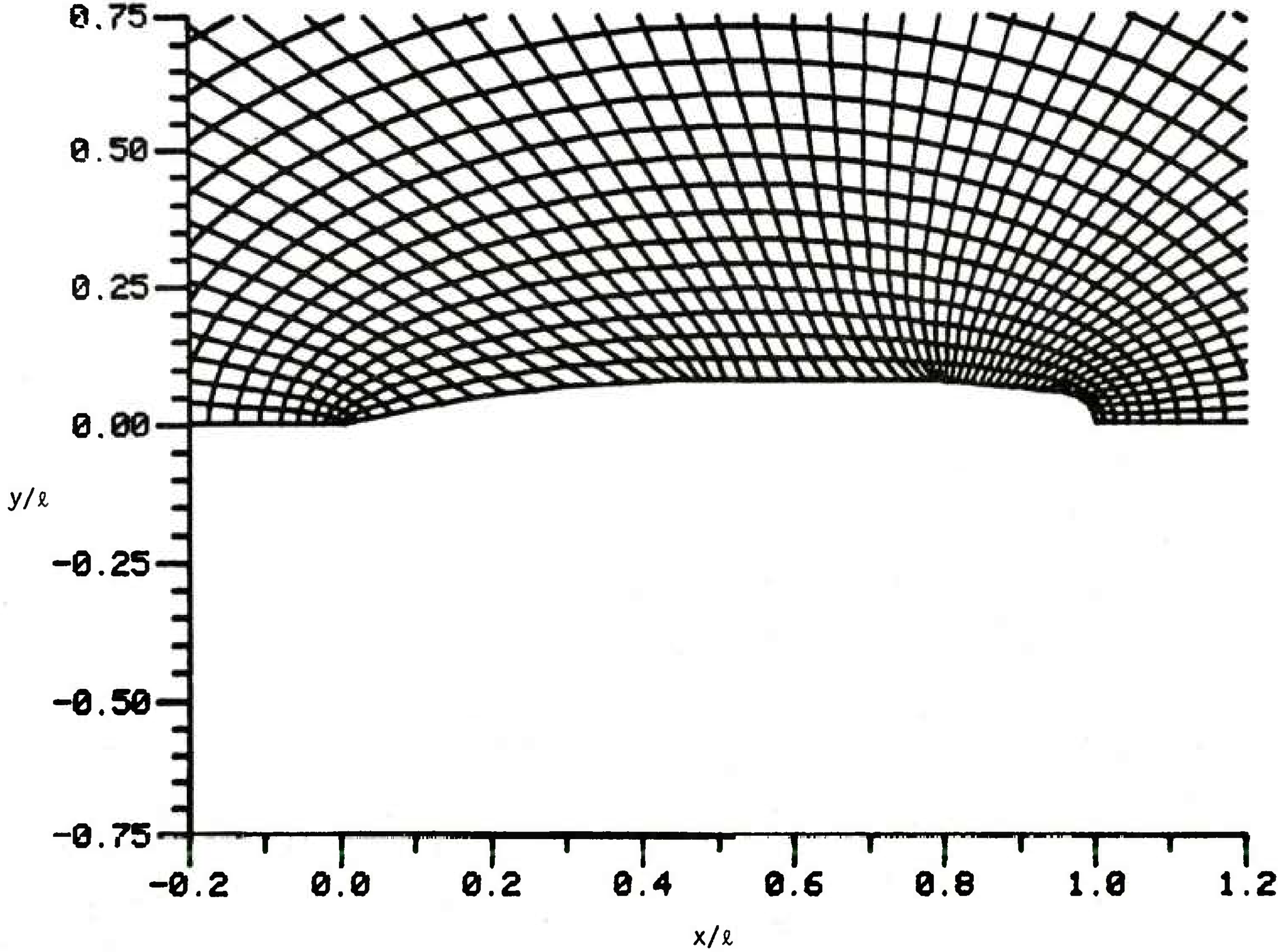


Figure 10b. Grid Detail near Projectile

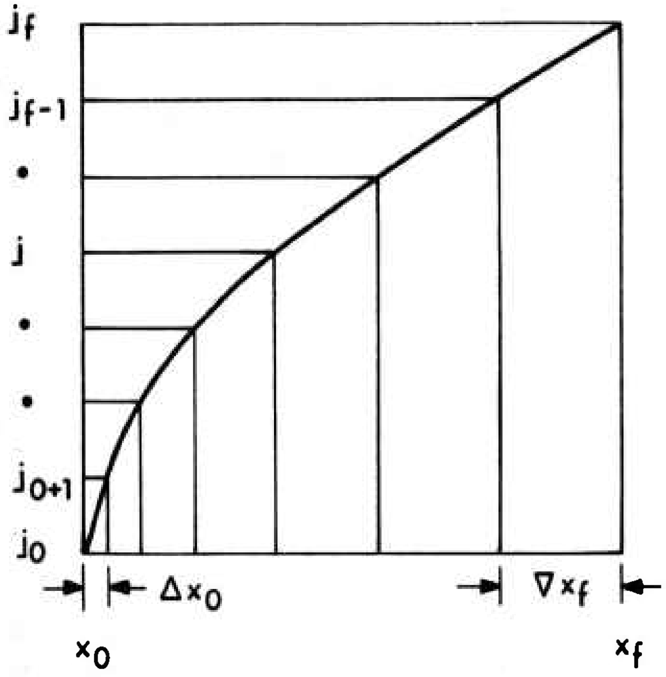
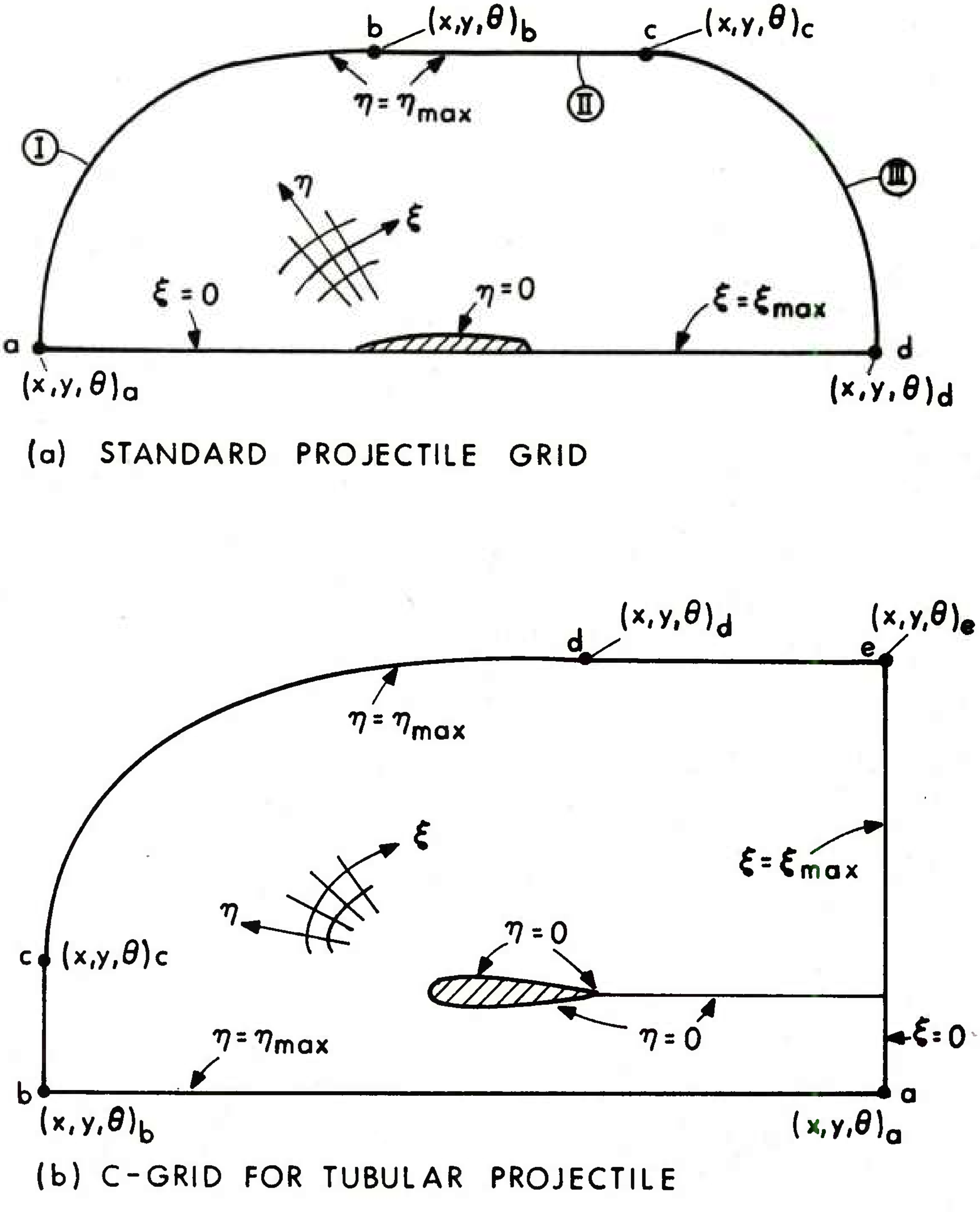


Figure 11. Stretching Function, Points j are Specified along with x_o, x_f, Δx_o, and ∇x_f



(a) STANDARD PROJECTILE GRID

(b) C-GRID FOR TUBULAR PROJECTILE

Figure 12. Outer Boundary Structure and Terminology for Two Classes of Grid

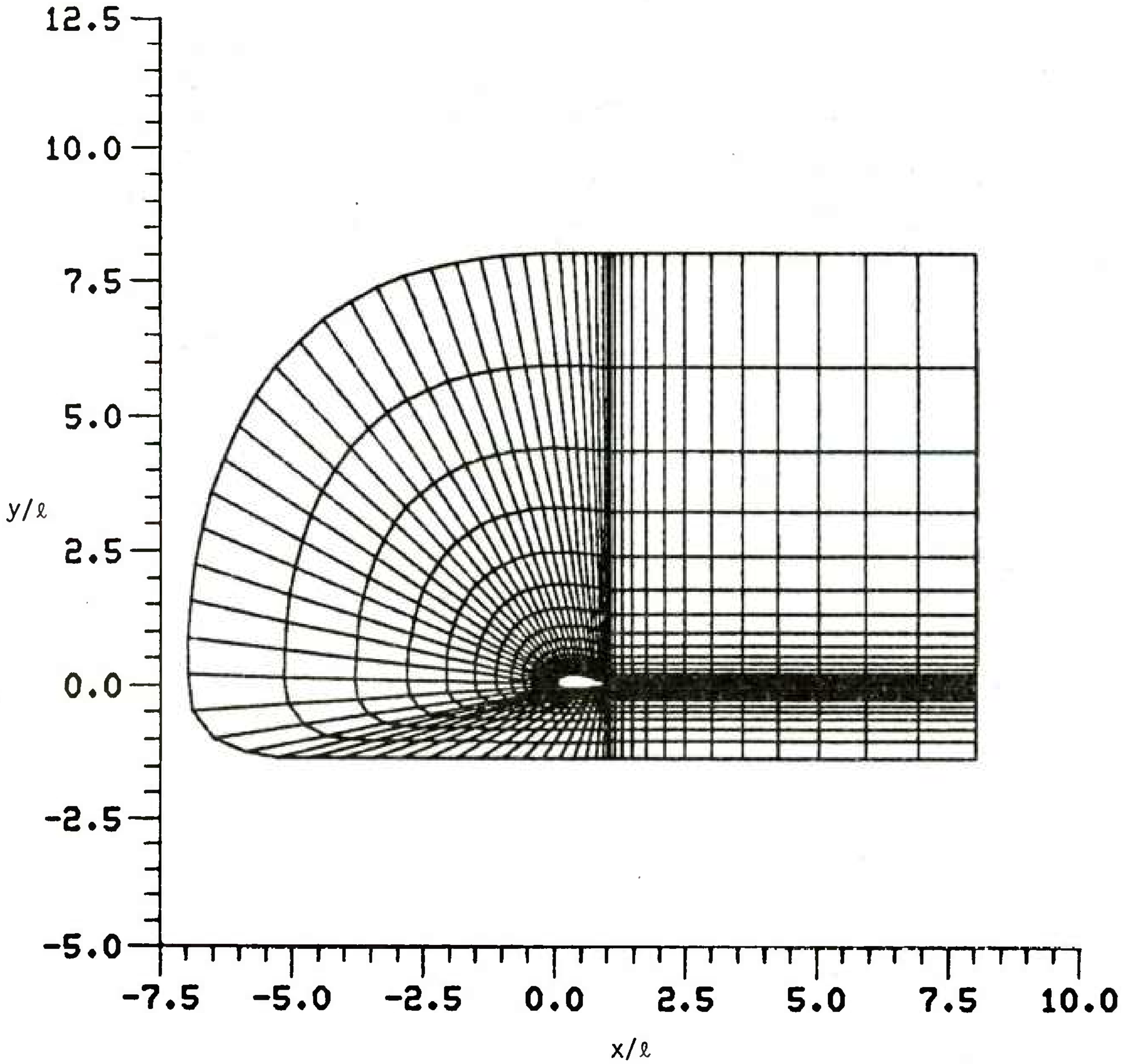


Figure 13a. Overview of Straight Ray Grid for Tubular Projectile

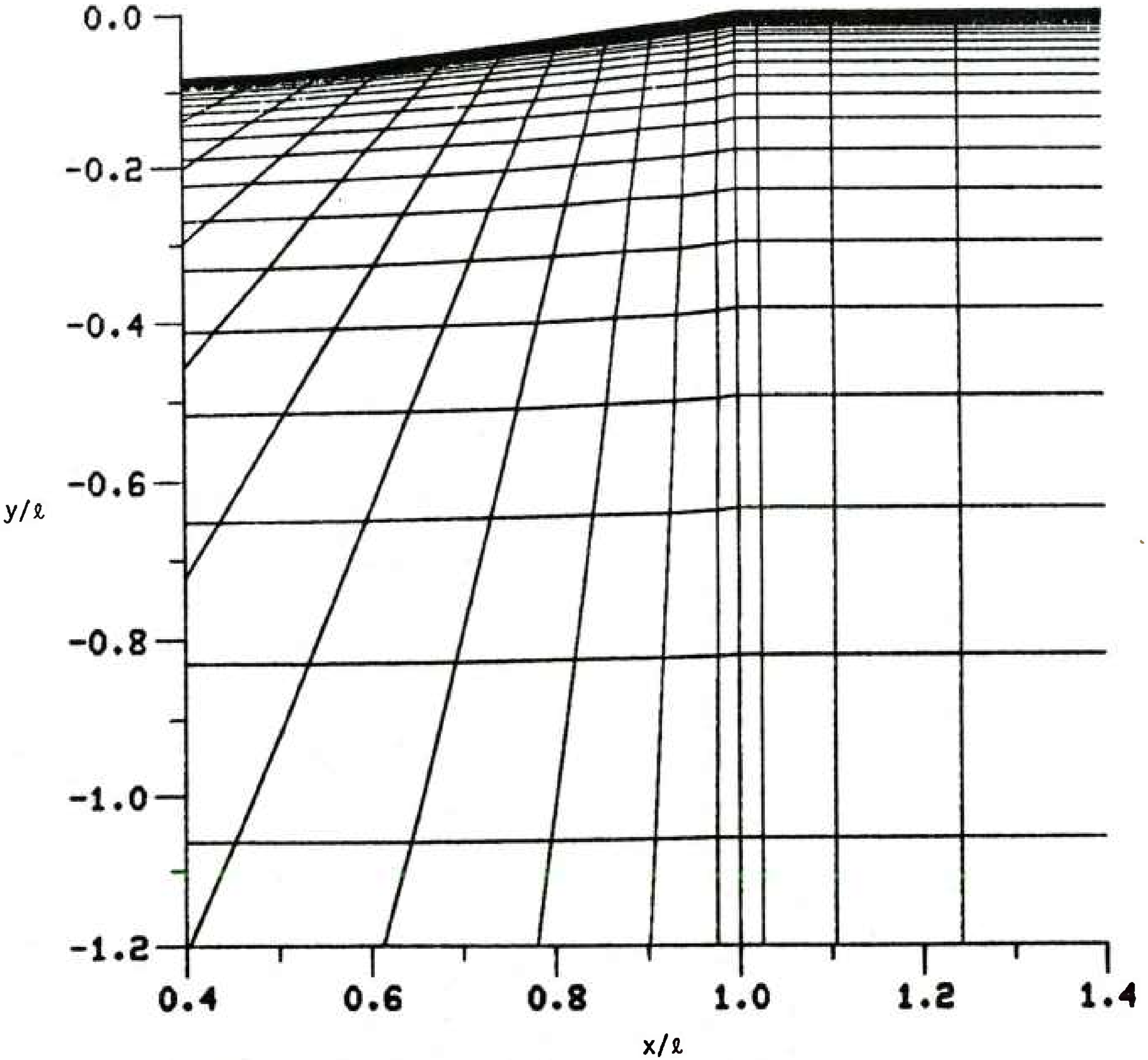


Figure 13b. Grid Detail near Lower Trailing Edge

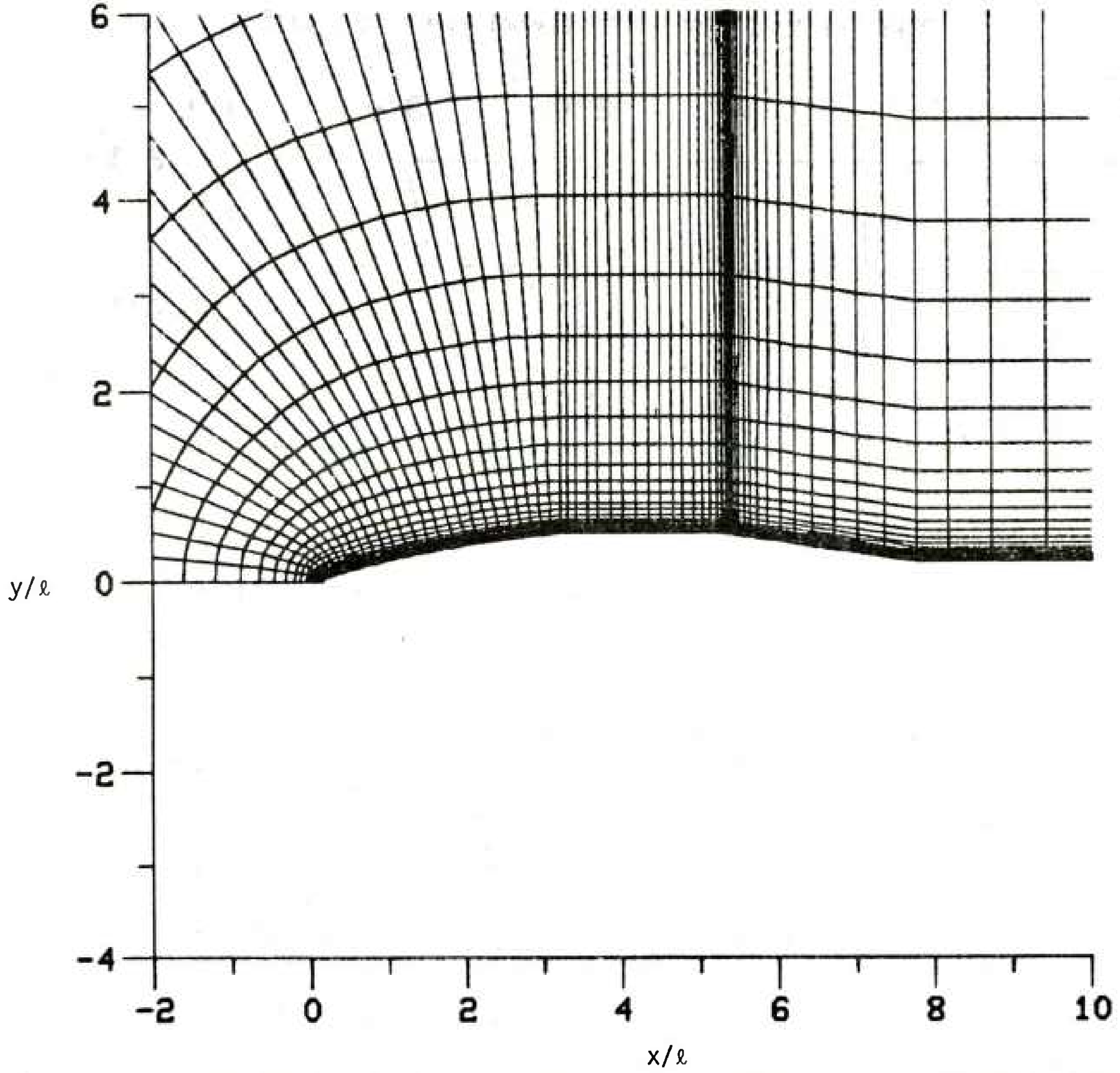


Figure 14a. Standard Projectile Grid with Controlled Reclustering along Lines of Constant ξ

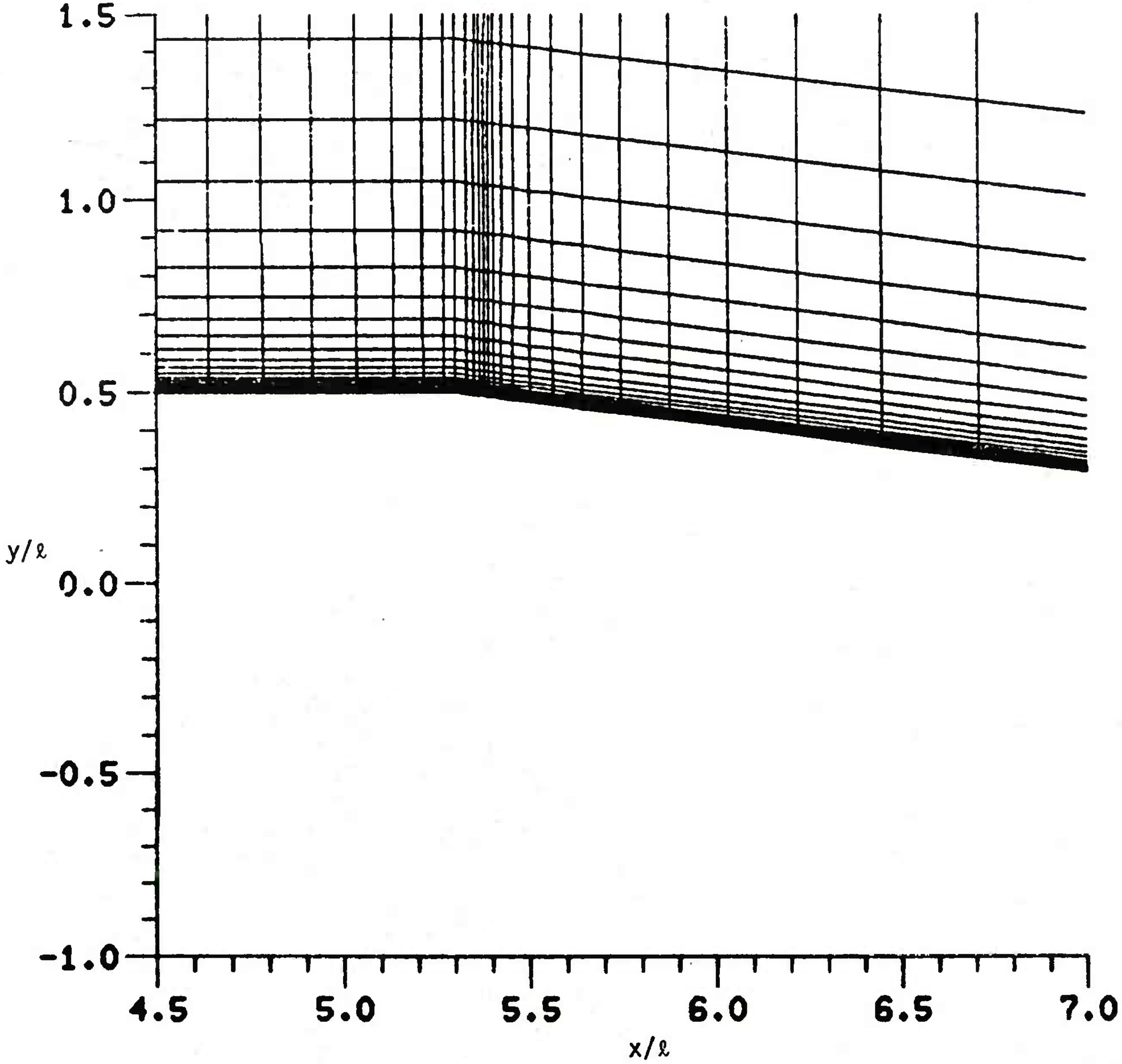


Figure 14b. Boattail Base Detail Showing Specified Clustering in η near the Body

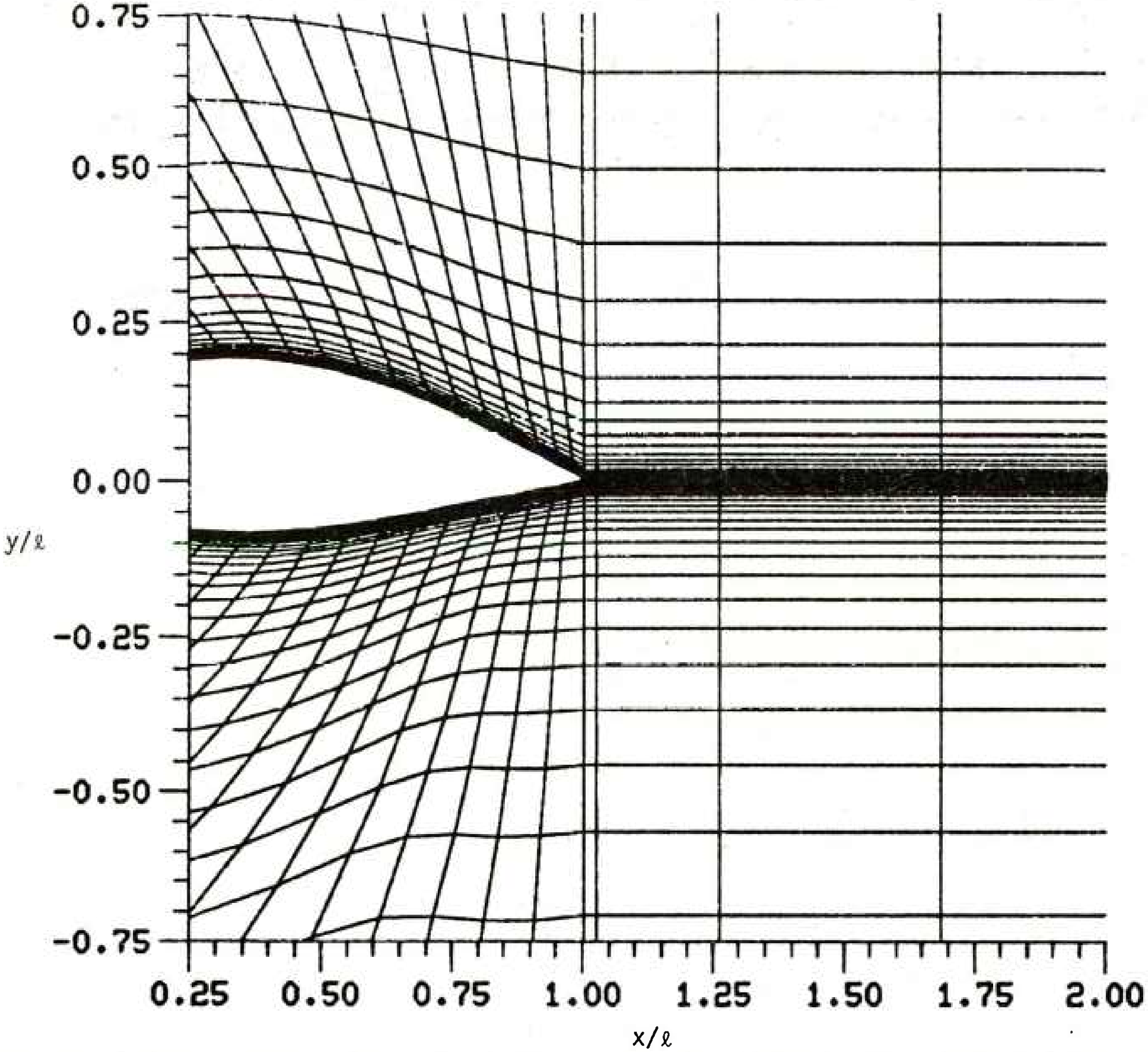


Figure 15. Hybrid Straight Ray and Elliptic Solver Grid Detail for a Tubular Projectile

LIST OF SYMBOLS

j	index value in ξ direction
k	index value in μ direction
s	arc length
x,y	physical cartesian coordinates
J	Jacobian of the transformation between the physical and the computational coordinates
P,Q	clustering parameters for the elliptical solver
Δ	forward finite difference
Δs	grid spacing at the inner boundary
∇	backward finite difference
ε	clustering parameter in η direction
θ	angle between segments of the outer boundary and the x-axis
ξ,η	computational coordinates in the axial and radial directions
ℓ	model length

Subscripts

f	final value
o	initial value

APPENDIX A

COMPUTER CODE DESCRIPTION AND LISTING

The computer program is a highly modular code. The main program is divided into three parts: inner boundary, outer boundary, and grid generation.

A. Inner Boundary

In forming the inner boundary, subroutine BODY is called to define the body shape and to distribute points along the body. Subroutine BODAN is called for an analytic shape. The user can modify this routine to supply his own body function. Subroutine BODAN calls subroutine BODIS which is the routine that clusters according to Eq. (1). If the body is not analytic, a table of x,y ordinates are read from BODY. These ordinates are normalized and then distributed as a function of axis length (chord) or arc length with calls to BODIS. The newly distributed body points are interpolated from the table of ordinates using cubic splines (subroutine CSPLIN). For example, ordinates of y versus arc length $\underline{s}$ are interpolated to form $\bar{y}$, as a function of the distribution arc length $\bar{s}$. Subroutine BODY then returns to main. At this point a sting and/or cut can be added by calls to STING (i.e., a sting as in Figure 1 or upper cut as in Figure 4) and CARTB (i.e., forward cut in Figure 2, lower cut in Figure 4).

Points are distributed along the sting and/or cut using BODIS. In subroutine STING, points are read in from 1 to NCGRD. NCGRD-1 points are added to the total count in ξ. Likewise in CARTB a set of points along the cut are added to the previous number. The final number of x,y inner boundary points is printed in main.

B. Outer Boundary

MAIN calls subroutine OUTER which forms the outer boundary. Here the cubic segments, as defined by Eq. (4), are read in and joined together. Allowance is made for 8 possible segments. Finally, points are distributed along this boundary as a function of arc length. Either a uniform distribution is used, or again subroutine BODIS is employed. Interpolation of the distributed points is again obtained by cubic splines, but the cubic spline function is restricted within an originally defined segment. Thus in Figure 12b the cubic spline interpolation is not carried from a to c, but is carried a to b, b to c, etc. In this way the discontinuous corner is not spline fit.

C. Grid Generation

Finally MAIN calls subroutine ALGRD. In subroutine ALGRD the straight line ray grid is formed using uniform clustering in η. Any segment of the grid between vertical or horizontal boundaries can then be regenerated using subroutine RELAX to obtain an SLOR solution to Eq. (9). Finally, the grid lines can be exponentially reclustered in η using Eq. (8). The grid is then stored for display or computational purposes.

D. Subsidiary Subroutines

With storage of the grid the program ends. Besides those subroutines described above, several other routines are called. Subroutine CLUST is called by BODIS and this is the routine that literally corresponds to the distribution function, Eq. (1). Subroutines TRIB and TRIP are routines for the solution of the tridiagonal matrix which must be inverted in the successive line overrelaxation procedure used by RELAX.

The TRIB routine is for conventional tridiagonal matrices, the TRIP routine is for periodic tridiagonal matrices. Finally, subroutine INIPQ is used to input $\overline{P}$ and $\overline{Q}$ of Eq. (9). Use of these terms is not currently recommended.

Two subroutines are called from BODAN to describe the blunt, secant-ogive, nose projectile. SCALC computes x-values associated with the nose cap. The fuse height is used to vary the degree of bluntness. SECANT is then called to provide the analytic functions used to compute the points along the remaining body configuration.

Subroutine GSPIN is called only when a three-dimensional grid is required. Both three- and two-dimensional grids are written, however, the former is used for flowfield computations and the latter is used for plotting.

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9,TAPE10)
C
C  CHECK FOR 3D GRID
C
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1  , T(100), TS(100)
      COMMON /COMP/ X(100), Y(100)
C
C
      READ (5,60) I3D,ND,LMAX
      WRITE (6,50) I3D,ND,LMAX
C
C
C   DISTRIBUTE POINTS ALONG INNER BOUNDARY
      WRITE (6,70)
      CALL BODY
      READ (5,60) NFLAG
      IF (NFLAG.LT.0) GO TO 40
      READ (5,60) NCGRD,NCART
      WRITE (6,80) NCGRD,NCART
      IF (NCGRD.GT.0) CALL STING (NCGRD)
      IF (NCART.GT.0) CALL CARTB (NCART)
      JMAX=JBOD
      JM=JMAX-1
      WRITE (6,90) JMAX
      IF (NCGRD.GT.0.OR.NCART.GT.0) GO TO 10
      GO TO 30
   10 WRITE (6,100)
      DO 20 J=1,JMAX
      WRITE (6,110) J,XX(J),YY(J)
   20 CONTINUE
   30 CONTINUE
C
C   FORM OUTER BOUNDARY
      WRITE (6,120)
      READ (5,60) NSEGS,IOUTO
      WRITE (6,130) JMAX,NSEGS
      CALL OUTER (NSEGS,IOUTO)
C
C   GRID GENERATION
      WRITE (6,140)
      CALL ALGRD (ISTOR)
C
C
C      FORM 3D GRID.   LMAX IS CIRCUMFERENTIAL DIRECTION
      IF (I3D.EQ.1) CALL GSPIN (I3D,ND,ISTOR,LMAX)
   40 STOP
C
   50 FORMAT (1H0,11HI3D,ND,LMAX,3I5)
   60 FORMAT (8I5)
   70 FORMAT (1H1,36H********** INNER BOUNDARY **********)
   80 FORMAT (1H0,13H NCGRD,NCART ,2I5)
   90 FORMAT (1H0,21H FINAL VALUE OF JMAX ,I5)
  100 FORMAT (1H0,43H FINAL VALUES OF J,X,Y ALONG INNER BOUNDARY)
  110 FORMAT (1H ,I5,2F13.6)
  120 FORMAT (1H1,37H ********** OUTER BOUNDARY **********)

  130 FORMAT (1H0,10HJMAX,NSEGS,2I5)
  140 FORMAT (1H1,39H ********** GRID GENERATION ********** )
      END
      SUBROUTINE ALGRD (ISTOR)
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /BODY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1  , T(100), TS(100)
      COMMON /GRID/ X(80,60), Y(80,60)
      COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)
C
C    FORM ALGEBRAIC GRID OR ELLIPTIC EQ. GENERATED GRID
C
      READ (5,150) KMAX,ITERM,IPER,NCLUS,ISTOR,JELLI
      WRITE (6,160) KMAX,ITERM,IPER,NCLUS,ISTOR,JELLI
      KM=KMAX-1
      READ (5,170) DS,OMEGA
      WRITE (6,180) DS,OMEGA
C
C    STRAIGHT RAY GRID USED IF ITERM .LE. 0
C                          OTHERWISE USED AS INITIAL GUESS
      DO 40 J=1,JMAX
      X0=XX(J)
      X1=XS(J)
      Y0=YY(J)
      Y1=YS(J)
      R=SQRT((X1-X0)**2+(Y1-Y0)**2)
      EPS=0.
      IF (ITERM.LT.0.) GO TO 10
      DD=R/(KMAX-1)
      GO TO 20
   10 EPS=EPSIL(R,0.,DS,KMAX,0.00002,20,J)
      DD=DS
   20 X(J,1)=X0
      Y(J,1)=Y0
      TR=0.
      DO 30 K=2,KMAX
      TR=TR+DD*(1.+EPS)**(K-2)
      TT=TR/R
      X(J,K)=X0+(X1-X0)*TT
      Y(J,K)=Y0+(Y1-Y0)*TT
   30 CONTINUE
   40 CONTINUE
C
      IF (ITERM.LT.0) GO TO 70
C
C    ELLIPTIC P.D.E. GRID GENRATION SCHEME
      IF (JELLI.GE.1) GO TO 50
      CALL RELAX (ITERM,IPER,1,JMAX,OMEGA)
      GO TO 70
   50 DO 60 LL=1,JELLI
      READ (5,150) JI,JF
      WRITE (6,190) JI,JF
   60 CALL RELAX (ITERM,IPER,JI,JF,OMEGA)
   70 CONTINUE
C
C    CLUSTERING OPTION
      IF (NCLUS.LT.0) GO TO 130
      DO 120 J=1,JMAX
      T(1)=0.
      DO 80 K=2,KMAX
```

```
 80 T(K)=T(K-1)+SQRT((X(J,K)-X(J,K-1))**2+(Y(J,K)-Y(J,K-1))**2)
    EPS=EPSIL(T(KMAX),0.,DS,KMAX,0.00002,20,J)
    S(1)=0.
    N=-1
    DO 90 K=2,KMAX
    N=N+1
 90 S(K)=S(K-1)+DS*(1.+EPS)**N
    DO 100 K=1,KMAX
    XX(K)=X(J,K)
100 YY(K)=Y(J,K)
    CALL CSPLIN (S,XS,T,XX,A,B,C,D,F,H,2,KM,1,KMAX)
    CALL CSPLIN (S,YS,T,YY,A,B,C,D,F,H,2,KM,1,KMAX)
    DO 110 K=2,KM
    X(J,K)=XS(K)
110 Y(J,K)=YS(K)
120 CONTINUE
C
130 CONTINUE
C
C  SURPRESS THE PRINTOUT
C     K = KMAX/2
C     DO 20 J=1,JMAX
C     WRITE(6,603) X(J,2),Y(J,2),X(J,3),Y(J,3),X(J,4),Y(J,4),
C    1  X(J,K),Y(J,K),X(J,KM),Y(J,KM)
C 603 FORMAT(1H ,10F12.5)
C  20 CONTINUE
C
C
C   OPTIONAL STORE OF DATA
      IF (ISTOR.LE.0) GO TO 140
      WRITE (9) ((X(J,K),J=1,JMAX),K=1,KMAX),((Y(J,K),J=1,JMAX),K=1,
     1  KMAX)
  140 CONTINUE
      RETURN
C
  150 FORMAT (8I5)
  160 FORMAT (1H0,' SUB. ALGRD PRINTOUT.... KMAX,ITERM,IPER,NCLUS,ISTOR,
     1 JELLI',/,8I5)
  170 FORMAT (8F10.0)
  180 FORMAT (1H0,11H DS, OMEGA ,2F13.5)
  190 FORMAT (1H0,26HELLIPTIC GRID FORMED FROM ,I3,5H  TO ,I3)
      END
      SUBROUTINE BODAN (ISEGS)
      COMMON /CALC/ XO, XATF, BS, ICT, FLAG, POYDX, FXF, RFN
      COMMON /INPS/ X1, X2, X3, X4, RAD, DYDX, CHORD, FUSE, AS, RADS
      COMMON JMAX, KMAX, JM, KM, NBDO, JBDD
      COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1  , T(100), TS(100)
C
      READ (5,100) TAU,FLAG
      WRITE (6,110) TAU,FLAG
      JWRIT=JBDD
      ICT=0
      IF (FLAG.GE.1.) CALL SCALC
      CALL BODIS (ISEGS,JWRIT)
      DO 10 J=1,JBDD
   10 XX(J)=S(J)
C
C    ANALYTIC BODY SHAPE
      IF (FLAG.GE.0.) GO TO 30
      DO 20 J=1,JBDD
      XX(J)=S(J)/S(JBDD)
C    PARABOLIC ARC
   20 YY(J)=2.*TAU*XX(J)*(1.-XX(J))
      GO TO 90
   30 CONTINUE
      IF (FLAG.LE.0.) GO TO 40
      CALL SECANT
      GO TO 90
C
C    PROJECTILE WITH TANGENT OGIVE,CYLINDER,BOATTAIL, CIRCULAR CAP
   40 READ (5,100) X1,X2,X3,X4,RAD,THETA,CHORD
      THETA=THETA*.0174533
      DYDX=TAN(THETA)
      BASE=RAD+(X4-X3)*DYDX
      XE=BASE*(DYDX+SQRT(1.+DYDX**2))
      X5=X4+XE
C    NONDIMENSIONAL OPTION
      IF (CHORD.GE.0.) CHORD=X5-X1
      CHORD=ABS(CHORD)
      WRITE (6,120) X1,X2,X3,X4,RAD,THETA,CHORD
      RCH=1./CHORD
      X1=X1*RCH
      X2=X2*RCH
      X3=X3*RCH
      X4=X4*RCH
      X5=X5*RCH
      BASE=BASE*RCH
      XE=XE*RCH
      RAD=RAD*RCH
      DO 80 J=1,JBDD
      IF (XX(J).GE.X2) GO TO 50
      XOG=X2-X1
      XBAR=(XX(J)-X1)/XOG
      VLAM=XOG/RAD
      VSQ=VLAM**2
      RBAR=VSQ+.25
      RADI=1.-VSQ*(1.-XBAR)**2/(RBAR**2)
      YY(J)=(1.-2.*RBAR*(1.-SQRT(RADI)))*RAD

      GO TO 80
   50 IF (XX(J).GE.X3) GO TO 60
      YY(J)=RAD
      GO TO 80
   60 IF (XX(J).GE.X4) GO TO 70
      YY(J)=RAD+(XX(J)-X3)*DYDX
      GO TO 80
   70 RS=(BASE**2)*(1.+DYDX**2)
      XBAR=(XX(J)-X4)
      RADI=RS-(XBAR-BASE*DYDX)**2
      YY(J)=SQRT(RADI)
   80 CONTINUE
C    END OF PROJECTILE
   90 CONTINUE
      RETURN
C
  100 FORMAT (8F10.0)
  110 FORMAT (1H0,8HTAU,FLAG,2F14.5)
  120 FORMAT (1H0,29H X1,X2,X3,X4,RAD,THETA,CHORD ,/,7F14.5)
      END
      SUBROUTINE BODIS (ISEGS,JWRIT)
      COMMON /CALC/ XO, XATF, BS, ICT, FLAG, PDYDX, FXF, RFN
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1  , T(100), TS(100)
C
C     S DISTRIBUTION ON BODY
C
      DO 10 I=1,ISEGS
      READ (5,20) JI,JF,XI,XF,DXI,DXF
      WRITE (6,30) JI,JF,XI,XF,DXI,DXF
      CALL CLUST (JI,JF,XI,XF,DXI,DXF,S)
   10 CONTINUE
      WRITE (6,40) (S(J),J=1,JWRIT)
      RETURN
C
   20 FORMAT (2I5,6F10.0)
   30 FORMAT (1H0,21H JI,JF,XI,XF,DXI,DXF ,2I5,4F12.5)
   40 FORMAT (1H ,10F11.5)
      END
      SUBROUTINE BODY
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1  , T(100), TS(100)
      COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)
      COMMON /COMP/ X(100), Y(100)
C
      READ (5,110) NBOD,JBOD,IXDRS,ISEGS
      WRITE (6,120) NBOD,JBOD,IXDRS,ISEGS
C   IF NBOD IS NEGATIVE, ANALYTIC SHAPE IS USED
      IF (NBOD.GE.0) GO TO 10
      NBOD=-NBOD
      CALL BODAN (ISEGS)
      GO TO 90
C
   10 CONTINUE
      WRITE (6,130)
      READ (5,150) CHORD
      WRITE (6,140) CHORD
C   READ CARDS IN CLOCKWISE NOSE TO TAIL
      DO 20 J=1,NBOD
      READ (5,150) X(J),Y(J)
      WRITE (6,160) J,X(J),Y(J)
      X(J)=X(J)/CHORD
      Y(J)=Y(J)/CHORD
   20 CONTINUE
C   COMPUTE NORMALIZED ARC LENGTH TO USE AS A MONOTONE PARAMETER
      SS(1)=0.
      DO 30 J=2,NBOD
   30 SS(J)=SS(J-1)+SQRT((X(J)-X(J-1))**2+(Y(J)-Y(J-1))**2)
      DO 40 J=2,NBOD
   40 SS(J)=SS(J)/SS(NBOD)
C
C   COMPUTE A NORMALIZED CLUSTERED PARAMETRIC FUNCTION FOR DISTRIBUTUON
C   OF BODY POINTS
      JWRIT=JBOD
      CALL BODIS (ISEGS,JWRIT)
      SAV=S(1)
      S(1)=0.
      DO 50 J=2,JBOD
      SDEL=ABS(S(J)-SAV)
      SAV=S(J)
   50 S(J)=S(J-1)+SDEL
      DO 60 J=1,JBOD
   60 S(J)=S(J)/S(JBOD)
C   OPTION WHETHER TO SET X EQUAL TO S DISTRIBUTION
      IF (IXDRS.LT.0) GO TO 70
      CALL CSPLIN (S,XX,SS,X,A,B,C,D,F,H,1,JBOD,1,NBOD)
      CALL CSPLIN (S,YY,SS,Y,A,B,C,D,F,H,1,JBOD,1,NBOD)
      GO TO 90
   70 CONTINUE
      DO 80 J=1,JBOD
   80 XX(J)=X(1)+S(J)*(X(NBOD)-X(1))
      CALL CSPLIN (XX,YY,X,Y,A,B,C,D,F,H,1,JBOD,1,NBOD)
C
   90 CONTINUE
      WRITE (6,170)
      DO 100 J=1,JBOD
      WRITE (6,160) J,XX(J),YY(J)
  100 CONTINUE
      RETURN
C
  110 FORMAT (4I5)
  120 FORMAT (1H0,22HNBOD,JBOD,IXDRS,ISEGS ,4I5)
  130 FORMAT (1H0,38H X,Y INPUT DEFINING BODY... SUB. BODY )
  140 FORMAT (1H0,32H NORMALIZING CHORD LENGTH INPUT ,F13.5)
  150 FORMAT (2F10.0)
  160 FORMAT (1H ,16H J,X,Y, ON BODY ,I5,2F14.5)
  170 FORMAT (1H1,32H J AND BODY DISTRIBUTED X AND Y )
      END
      SUBROUTINE GSPIN (I3D,ND,ISTOR,LMAX)
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /GRID/ X(80,60), Y(80,60)
      COMMON /GRID30/ X3(420,60), Y3(420,60), Z3(420,60)
      LEVEL 2,X3,Y3,Z3
      PI=4.*ARCTAN(1.)
      DS=PI/(LMAX-3)
      N1=KMAX*LMAX
      DO 10 K=1,KMAX
      DT=-2.*DS
      DO 10 L=1,LMAX
      DT=DT+DS
      KL=(K-1)*ND+L
      DO 10 J=1,JMAX
      Y3(KL,J)=Y(J,K)*SIN(DT)
      Z3(KL,J)=Y(J,K)*COS(DT)
      X3(KL,J)=X(J,K)
   10 CONTINUE
      IF (ISTOR.LE.0) GO TO 40
      REWIND 9
      WRITE (9) ((X3(KL,J),KL=1,N1),J=1,JMAX),((Y3(KL,J),KL=1,N1),J=1,
     1  JMAX),((Z3(KL,J),KL=1,N1),J=1,JMAX)
C
C   REWRITE 3D DATA FOR 2D PLOTTING
C
      DO 30 J=1,JMAX
      DO 20 KL=2,N1,LMAX
      N=1+(KL-2)/ND
      X(J,N)=X3(KL,J)
      Y(J,N)=Z3(KL,J)
   20 CONTINUE
   30 CONTINUE
      WRITE (10) ((X(J,N),J=1,JMAX),N=1,KMAX),((Y(J,N),J=1,JMAX),N=1,
     1  KMAX)
C
   40 CONTINUE
      WRITE (6,90)
      DO 60 L=2,LMAX,3
      DO 60 K=1,KMAX,4
      KL=(K-1)*ND+L
      WRITE (6,70) L,K,L,K+1
      KL2=K*ND+L
      DO 50 J=1,JMAX,2
   50 WRITE (6,80) J,X3(KL,J),Y3(KL,J),Z3(KL,J),J,X3(KL2,J),Y3(KL2,J),
     1  Z3(KL2,J)
   60 CONTINUE
      RETURN
C
   70 FORMAT (1H0,5H  J  ,2HK=,I2,3H L=,I2,9X,1HX,10X,1HY,12X,1HZ,13X,
     1 3HJ  ,2HK=,I2,3H L=,I2,5X,1HX,10X,1HY,11X,1HZ)
   80 FORMAT (1H ,I3,14X,F10.5,3X,F10.5,3X,F10.5,9X,I2,14X,F10.5,2X,F10.
     15,2X,F10.5)
   90 FORMAT (1H1)
      END
      SUBROUTINE OUTER (NSEGS,IOUTO)
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1  , T(100), TS(100)
      COMMON /COMP/ X(100), Y(100)
      COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)
C
C   THIS PROGRAM FORMS AN OUTER GRID BOUNDARY USING CONTIGUOUS CUBIC
C   SEGMENTS. NUMBER OF SEGMENTS IS NSEGS.  POINT AND SLOPE ARE INPUT AT
C   THE ENDS OF A SEGMENT.  SLOPE IS AN ANGLE IN DEGRESS. PARAMETRIC
C   CUBICS  USED TO PERMIT ANY SLOPE  (THETA = 90, -90, ETC). INITIAL
C   LOGIC DETERMINES CUBIC COEFFICIENTS OF EACH SEGMENT. REMAINING
C   LOGIC DISTRIBUTES POINTS ALONG OUTER BOUNDARY USING ARC LENGTH
C   AS DISTRIBUTION FUNCTION. THUS TWO PARAMETRIC VARIABLES ARE USED.
C   FINDING X,Y SO CUBIC SEGMENTS CAN BE DISJOINT IN SLOPE IS MESSY
C   A SINGLE SPLINE INTERPOLATION CANNOT BE USED OVER THE COMBINED
C   SEGMENTS BECAUSE OF POSSIBLE SLOPE DISCONTINUITY.
C
      DIMENSION JA(8), JB(8)
      DIMENSION CA0(8), CA1(8), CA2(8), CB0(8), CB1(8), CB2(8), CARC(8)
      DO 110 N=1,NSEGS
C   POINTS AND SLOPES,  -90 .LE.  THETA  .LE. 90    DEGREES USED
      READ (5,240) X0,Y0,X1,Y1,TH0,TH1
      WRITE (6,250) X0,Y0,X1,Y1,TH0,TH1
      RTH0=0.017453292*TH0
      RTH1=0.017453292*TH1
      XI=X1-X0
      ETA=Y1-Y0
C
C   SET MFLAG,  LOGIC CHIEFLY USED TO AVOID INFINITE DY/DX, USES DX/DY=0
      TA=SIN(RTH0)
      TB=COS(RTH0)
      TC=SIN(RTH1)
      TD=COS(RTH1)
      IF (ABS(TA).GT.ABS(TB)) GO TO 10
      MFLAG=1
      IF (ABS(TC).GT.ABS(TD)) MFLAG=2
      GO TO 20
   10 MFLAG=3
      IF (ABS(TC).GT.ABS(TD)) MFLAG=4
   20 CONTINUE
C   DETERMINE COEFFICIENTS FOR PARAMETRIC CUBICS
C   SET UP LINEAR COEFFS. FIRST. INDEFINITE CUBIC DEFAULTS TO LINEAR
      A1=XI
      A2=0.
      B1=ETA
      B2=0.
C
      GO TO (30,40,50,60), MFLAG
   30 DYDX0=TA/TB
      DYDX1=TC/TD
      TEST=DYDX1-DYDX0
      IF (ABS(TEST).LT.0.0005) GO TO 80
      SM1=2.*(ETA-DYDX0*XI)/TEST
      SM0=2.*XI-SM1
      SN0=SM0*DYDX0
      GO TO 70
   40 OYOXO=TA/T8
      OXOY1=TO/TC
      TEST=1.-OYOXO*OXOY1
      IF (ABS(TEST).LT.0.0005) GO TO 80
      SMO=2.*(XI-OXOY1*ETA)/TEST
      SNO=SMO*OYOXO
      GO TO 70
   50 OXOYO=T8/TA
      OYOX1=TC/TO
      TEST=1.-OYOX1*OXOYO
      IF (ABS(TEST).LT.0.0005) GO TO 80
      SNO=2.*(ETA-DYOX1*XI)/TEST
      SMO=SNO*OXOYO
      GO TO 70
   60 DXOYO=T8/TA
      OXOY1=TO/TC
      TEST=OXOY1-OXOYO
      IF (ABS(TEST).LT.0.0005) GO TO 80
      SN1=2.*(XI-OXOYO*ETA)/TEST
      SNO=2.*ETA-SN1
      SMO=SNO*OXOYO
C
   70 A1=SMO
      A2=XI-SMO
      81=SNO
      82=ETA-SNO
   80 CONTINUE
      JN8R=25
C
C   COMPUTE NUMERICAL ARC LENGTH AS A PARAMETER( EXACT ARC LENGTH IS
C        POSSIBLE BUT INVERSE PROCESS IS NOT
      00 90 J=1,JN8R
      TT=JN8R-1
      TT=(J-1)/TT
      X(J)=XO+TT*(A1+A2*TT)
   90 Y(J)=YO+TT*(81+82*TT)
      SARC=0.
C
C   NOTE ...COULO USE SUM OF SQUARES AS PARAMETER RATHER THAN ARC LENGTH
C   IN THIS WAY ONE CAN AVOIO SQUARE ROOT CALCULATION   MUST USE EVERY
C     WHERE
C
      00 100 J=2,JN8R
  100 SARC=SARC+SORT((X(J)-X(J-1))**2+(Y(J)-Y(J-1))**2)
C   OATA FOR EACH CUBIC SEGMENT
      CA1(N)=A1
      CA2(N)=A2
      C81(N)=81
      C82(N)=82
      CAO(N)=XO
      C8O(N)=YO
      CARC(N)=SARC
      WRITE (6,260) XO,A1,A2,YO,81,82,SARC
C
  110 CONTINUE
C
C   CUBICS OETERMINEO, NOW OISTRIBUTE POINTS
```

```
C
C   TOTAL OUTER ARC LENGTH
      SARC=0.
      DO 120 N=1,NSEGS
  120 SARC=SARC+CARC(N)
      WRITE (6,270) SARC
C
C   DEFINE A UNIFORM OUTER DISTRIBUTION ARC LENGTH
      RH=1./(JMAX-1)
      DO 130 J=1,JMAX
  130 SS(J)=(J-1)*RH
C
C   OPTIONAL USE OF CUBIC SEGMENTS TO CLUSTER
      IF (IOUTO.LE.0) GO TO 150
      CALL BODIS (IOUTO,JMAX)
      DO 140 J=1,JMAX
  140 SS(J)=(S(J)-S(1))/(S(JMAX)-S(1))
  150 CONTINUE
      WRITE (6,280) (SS(J),J=1,JMAX)
C   NORMALIZE OUTER ARC LENGTH SEGMENTS TO SCALE OF DISTRIBUTION ARC LEN
      CA=0.
      DO 160 N=1,NSEGS
      CA=CA+CARC(N)/SARC
  160 CARC(N)=CA
      WRITE (6,290) (CARC(N),N=1,NSEGS)
C   FIND J INDICES  LIMITS WITHIN A SEGMENT
      N=1
      JA(N)=1
      DO 180 J=2,JMAX
      IF (SS(J).LE.CARC(N)) GO TO 170
      N=N+1
      JA(N)=J
  170 JB(N)=J
  180 CONTINUE
      DO 190 N=1,NSEGS
      WRITE (6,300) JA(N),JB(N)
  190 CONTINUE
C
C   FORM PARAMETRIC ARRAYS,FROM DISTRIBUTED PARAMETRIC ARRAY,
C   USE IT TO DETERMINE X,Y WITHIN A OUTER SEGMENT CURVE.
C   SPLINE REQUIRES ABOUT 5 POINTS IN AN INTERVAL
      S(1)=0.
      RT=1./(JNBR-1)
      DO 220 N=1,NSEGS
      T(1)=0.
      IF (N.GT.1) S(1)=CARC(N-1)
      X(1)=CAO(N)
      Y(1)=CBO(N)
      DO 200 J=2,JNBR
      TT=(J-1)*RT
      T(J)=TT
      X(J)=CAO(N)+TT*(CA1(N)+TT*CA2(N))
      Y(J)=CBO(N)+TT*(CB1(N)+TT*CB2(N))
      DS=SQRT((X(J)-X(J-1))**2+(Y(J)-Y(J-1))**2)
      S(J)=S(J-1)+DS/SARC
      WRITE (6,280) T(J),X(J),Y(J),S(J)
  200 CONTINUE
```

```
C
      J1=JA(N)
      J2=JB(N)
      CALL CSPLIN (SS,TS,S,T,A,B,C,D,F,H,J1,J2,1,JNBR)
      DO 210 J=J1,J2
      TT=TS(J)
      XS(J)=CA0(N)+TT*(CA1(N)+TT*CA2(N))
      YS(J)=CB0(N)+TT*(CB1(N)+TT*CB2(N))
      WRITE (6,280) TS(J),XS(J),YS(J)
  210 CONTINUE
  220 CONTINUE
      DO 230 J=1,JMAX
      WRITE (6,310) XS(J),YS(J)
  230 CONTINUE
      RETURN
C
  240 FORMAT (8F10.0)
  250 FORMAT (1H0,21H X0,Y0,X1,Y1,TH0,TH1 ,6F13.5)
  260 FORMAT (1H0,23HX0,A1,A2,Y0,B1,B2,SARC ,/,7F13.5)
  270 FORMAT (1H0,42H OUTER BOUNDARY NONDIMENSIONAL ARC LENGTH ,F14.6)
  280 FORMAT (1H ,8F12.5)
  290 FORMAT (1H0,6H CARC ,6F13.5)
  300 FORMAT (1H ,7H JA,JB ,2I5)
  310 FORMAT (1H ,6HXS,YS ,2F15.6)
      END
      SUBROUTINE RELAX (ITERM,IPER,JI,JF,OMEGA)
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /GRID/ X(80,60), Y(80,60)
      COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)
      DIMENSION IP(142), IR(142)
      DIMENSION G(100)
      COMMON /SOURCE/ P(80,2), Q(80,2), PFAC(2), QFAC(2)
C
C   SLOR SOLUTION OF ELLIPTIC GRID GENERATION EQS.
C DEL XI AND DEL ETA = 1.0
C
      J1=1
      J2=JMAX
      IF (IPER.GT.0) GO TO 10
      J1=JI+1
      J2=JF-1
   10 CONTINUE
      CALL INIPQ
C
      ITER=0
      KM=KMAX-1
C   SET PERIODIC INDICES
      DO 20 J=1,JMAX
      IP(J)=J+1
   20 IR(J)=J-1
      IP(JMAX)=1
      IR(1)=JMAX
C
C   FORM DIFFERENCE EXPRESSIONS AND TRIDIAGONALS
   30 ITER=ITER+1
      RSUM=0.
      DO 160 KK=2,KM
      K=KM+2-KK
      KP=K+1
      KR=K-1
      CP1=EXP((1-K)*PFAC(1))
      CP2=EXP((K-KMAX)*PFAC(2))
      CQ1=EXP((1-K)*QFAC(1))
      CQ2=EXP((K-KMAX)*QFAC(2))
      DO 40 J=J1,J2
      JP=IP(J)
      JR=IR(J)
      XXD=(X(JP,K)-X(JR,K))*.5
      XED=(X(J,KP)-X(J,KR))*.5
      YXD=(Y(JP,K)-Y(JR,K))*.5
      YED=(Y(J,KP)-Y(J,KR))*.5
      AD=XED**2+YED**2
      BD=-2.*(XXD*XED+YXD*YED)
      GD=XXD**2+YXD**2
      XXED=.25*(X(JP,KP)-X(JP,KR)-X(JR,KP)+X(JR,KR))
      YXED=.25*(Y(JP,KP)-Y(JP,KR)-Y(JR,KP)+Y(JR,KR))
      A(J)=AD
      B(J)=-AD-AD-GD-GD
      C(J)=AD
      F(J)=-BD*XXED-GD*(X(J,KP)+X(J,KR))
      G(J)=-BD*YXED-GD*(Y(J,KP)+Y(J,KR))
C   SOURCE TERMS
```

```
      DJAC=(XXD*YED-XED*YXD)
      DSQ=DJAC**2
      COFA=DSQ*(CP1*P(J,1)+CP2*P(J,2))
      COFB=DSQ*(CQ1*Q(J,1)+CQ2*Q(J,2))
      FOBX=SIGN(.5,COFA)
      FOBE=SIGN(.5,COFB)
      A(J)=A(J)-COFA*(.5-FOBX)
      B(J)=B(J)-2.*(COFA*FOBX+COFB*FOBE)
      C(J)=C(J)+COFA*(.5+FOBX)
      F(J)=F(J)-COFB*((.5+FOBE)*X(J,KP)-(.5-FOBE)*X(J,KR))
      G(J)=G(J)-COFB*((.5+FOBE)*Y(J,KP)-(.5-FOBE)*Y(J,KR))
   40 CONTINUE
C
      IF (IPER.GT.0) GO TO 130
C    SET B.C. AND INVERT
C
C    XI MIN AND MAX PLANES MUST BE X OR Y CARTESIAN PLANES
C
C    OUTFLOW B C ON X
C    TEST WHETHER XI PLANE IS X OR Y = CONSTANT PLANE
      IF (ABS(Y(JF,KMAX)-Y(JF,1)).LT.0.001) GO TO 50
C    OUTFLOW XI-PLANE TAKEN AS X=CONSTANT PLANE
      A(JF)=0.
      B(JF)=-1.
      C(JF)=0.
      F(JF)=-X(JF,1)
      GO TO 60
C    OUTFLOW XI-PLANE TAKEN AS Y= CONSTANT
   50 A(JF)=1.
      B(JF)=-1.
      C(JF)=0.
      F(JF)=(-X(JF-1,K)+X(JF-2,K))/3.
   60 CONTINUE
C    INFLOW B C ON X
      IF (ABS(Y(JI,KMAX)-Y(JI,1)).LT.0.001) GO TO 70
C    INFLOW XI-PLANE TAKEN AS X = CONSTANT PLANE
      A(JI)=0.
      B(JI)=-1.
      C(JI)=0.
      F(JI)=-X(JI,1)
      GO TO 80
   70 CONTINUE
C    INFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE
      A(JI)=0.
      B(JI)=-1.
      C(JI)=1.
      F(JI)=(-X(JI+1,K)+X(JI+2,K))/3.
   80 CALL TRIB (A,B,C,D,F,JI,JF)
C
C    Y B.C.AND INVERSION
      IF (ABS(Y(JF,KMAX)-Y(JF,1)).LT.0.001) GO TO 90
C    OUTFLOW XI-PLANE TAKEN AS X=CONSTANT PLANE
      A(JF)=1.
      B(JF)=-1.
      C(JF)=0.
      G(JF)=(-Y(JF-1,K)+Y(JF-2,K))/3.
      GO TO 100
```

```
C   OUTFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE
   90 A(JF)=0.
      B(JF)=-1.
      C(JF)=0.
      G(JF)=-Y(JF,1)
C   INFLOW
  100 IF (ABS(Y(JI,KMAX)-Y(JI,1)).LT.0.001) GO TO 110
C   INFLOW XI-PLANE TAXEN AS X = CONSTANT PLANE
      A(JI)=0.
      B(JI)=-1.
      C(JI)=1.
      G(JI)=-(Y(JI+1,K)-Y(JI+2,K))/3.
      GO TO 120
C   INFLOW XI-PLANE TAXEN AS Y = CONSTANT PLANE
  110 A(JI)=0.
      B(JI)=-1.
      C(JI)=0.
      G(JI)=-Y(JI,1)
  120 CALL TRIB (A,B,C,D,G,JI,JF)
      GO TO 140
C
  130 CONTINUE
C
C  PERIODIC B.C.
      CALL TRIP (A,B,C,F,D,H,1,JMAX)
      CALL TRIP (A,B,C,G,D,H,1,JMAX)
  140 CONTINUE
C
C   RELAXATION UPDATE
      DO 150 J=JI,JF
      YC=G(J)-Y(J,X)
      XC=F(J)-X(J,K)
      X(J,K)=X(J,X)+OMEGA*XC
      Y(J,K)=Y(J,K)+OMEGA*YC
  150 RSUM=RSUM+ABS(XC)+ABS(YC)
  160 CONTINUE
C
      IF ((ITER/10)*10.LT.ITER) GO TO 170
      WRITE (6,180) ITER,RSUM
  170 CONTINUE
      IF (ITER.LT.ITERM) GO TO 30
      RETURN
C
  180 FORMAT (1H ,26H ITERATION NBR AND RSUM    ,I5,E12.5)
      END
      SUBROUTINE INIPQ
      COMMON JMAX, KMAX, JM, KM, N800, J800
      COMMON /SOURCE/ P(80,2), Q(80,2), PFAC(2), QFACI2)
      DIMENSION PC(2), QC(2)
C
      READ (5,20) PFAC(1),QFAC(1),PFAC(2),QFAC(2)
      READ (5,20) PC(1),QC(1),PC(2),QC(2)
      WRITE (6,30) PFAC(1),QFAC(1),PFAC(2),QFACI2)
      WRITE (6,40) PC(1),QC(1),PC(2),QC(2)
C
      DO 10 N=1,2
      DO 10 J=1,JMAX
      P(J,N)=PC(N)
   10 Q(J,N)=QC(N)
      RETURN
C
   20 FORMAT (8F10.0)
   30 FORMAT (1H0,' EXPONENT COEFFICIENTS FOR SOURCE TERMS,PFAC,QFAC
     1 ',/,4F13.5)
   40 FORMAT (1H0,18H PC1,QC1,PC2,QC2  ,4F13.5)
      END
C
C
      SUBROUTINE CLUST (JI,JF,XI,XF,DXI,DXF,S)
      DIMENSION S(1)
C
      XFXI=XF-XI
      H=1./(JF-JI)
      H2=H*H
      H3=H2*H
      C=(DXF+DXI-2.*H*XFXI)/(H-3.*H2+2.*H3)
      B=(DXI-H*XFXI-C*(H3-H))/(H2-H)
      A=XFXI-B-C
C
      DO 10 J=JI,JF
      X=(J-JI)*H
   10 S(J)=XI+X*(A+X*(B+C*X))
      RETURN
      END
```

```
C
      FUNCTION EPSIL (FMX,FMIN,OFM,NPT,FPCC,ICC,NCALL)
C
C          THIS SUBROUTINE APPLIES A NEWTON-RAPHSON ROOT-FINDING
C          TECHNIQUE TO FIND A VALUE OF EPSILON FOR A PARTICULAR USE
C          OF THE EXPONENTIAL STRECHING TRANSFORMATION.
C
C    FMX IS TOTAL ARC LENGTH ALONG COORDINATE
C    FMIN IS STARTING VALUE OF ARC LENGTH   SUCH AS 0.0
C    OFM IS SPECIFIED INITIAL INCREMENT OF ARC LENGTH
C    NPT IS NUMBER OF POINTS ALONG COORDINATE
C    FPCC IS ITERATIVE ERROR BOUND, E.G.O( 0.00002)
C    ICC IS MAXIMUM NUMBER OF ITERATIONS
C    NCALL     IF NCALL=1 INITIAL GUESS FOR EPS IS USED
C              IF NCALL .GT. 1, PREVIOUS EPS USED AS INITIAL GUESS
C
C
      FMXL=FMK
      FMINL=FMIN
      OFML=OFM
      FPCCL=FPCC
      ICCL=ICC
C
C
      FNPTM2=NPT-2
      IF (NCALL.EQ.1) EPS=(FMKL/OFML)**(1.0/FNPTM2)-1.0
C
      DO 10 NIT=1,ICCL
      EP1=EPS+1.0
      EP1TN=EP1**FNPTM2
      REPS=1.0/EPS
      DFMOE=DFML*REPS
      F=FMXL-FMINL-OFMOE*(EP1TN*EP1-1.0)
      IF (ABS(F).LT.FPCCL) GO TO 20
      OFMOE2=OFMOE*REPS
      FPN=OFMOE2*(1.0+EP1TN*(EPS*FNPTM2-1.0))
      EPS=EPS+F/FPN
   10 CONTINUE
C
      EPSIL=EPS
      WRITE (6,30)
      RETURN
C
   20 EPSIL=EPS
C   SURPRESS THE ESPIL PRINTOUT
C     WRITE(6,601)  EPSIL,F,NIT
C
      RETURN
C
C
   30 FORMAT (/42H EKCEEDED MAX. NO. OF ITERATIONS IN EPSIL.)
      END
```

```
C
      SUBROUTINE CSPLIN (XX,YY,X,Y,A,B,C,D,F,H,N1,N2,J1,J2)
      DIMENSION XX(1), YY(1), X(1), Y(1), A(1), B(1), C(1), D(1), F(1),
     1 H(1)
C
C            CUBIC SPLINE INTERPOLATION
C  X,Y ARRAYS ARE TO BE INTERPOLATED
C  YY ARE FOUND INTERPOLATES CORRESPONDING TO XX
C  J1,,J2, ARE INDICE LIMITS ON X,Y
C  N1, N2 ARE INDICE LIMITS ON XX ( ALSO YY)
C  DIMENSION OF ARRAYS CARRIED IN FROM OUTSIDE, X(J) MUST BE MONOTONE
C FORMULA FROM NUMERICAL METHODS BY DAHLQUIST,BJORCK,ANDERSON
C                                                JLS FEB. 77
C
C      ROUNDING ERROR PROTECTION     CAUTION MAY MASK ERROR IN LOGIC
      IF (XX(N1).LT.X(J1)) XX(N1)=X(J1)
      IF (XX(N2).GT.X(J2)) XX(N2)=X(J2)
C
C
C       FIRST FIND DERIVATIVE LIKE TERMS THAT ARE COEFFICIENTS
      JA=J1+1
      JB=J2-1
      DO 10 J=JA,J2
      H(J)=X(J)-X(J-1)
   10 D(J)=(Y(J)-Y(J-1))/H(J)
      DO 20 J=JA,JB
      A(J)=H(J+1)
      B(J)=2.*(H(J)+H(J+1))
   20 F(J)=3.*(H(J)*D(J+1)+H(J+1)*D(J))
      B(J1)=2.
      H(J1)=1.
      F(J1)=3.*D(JA)
      A(J2)=1.
      B(J2)=2.
      F(J2)=3.*D(J2)
      CALL TRIB (A,B,H,C,F,J1,J2)
C
C  INTERPOLATION , X(J) ARRAY MUST BE MONOTONE
      J=J1
      I=J1+1
      DO 80 N=N1,N2
   30 IF (X(J).LE.XX(N).AND.X(I).GE.XX(N)) GO TO 70
      IF (X(I)-XX(N)) 40,40,50
   40 J=J+1
      I=I+1
      IF (I.GT.J2) GO TO 60
      GO TO 30
   50 J=J-1
      I=I-1
      IF (J.LT.J1) GO TO 60
      GO TO 30
   60 WRITE (6,90)
      STOP
   70 T=(XX(N)-X(J))/H(I)
      TT=1.-T
      YY(N)=T*Y(I)+TT*Y(J)+H(I)*T*TT*((F(J)-D(I))*TT-(F(I)-D(I))*T)
   80 CONTINUE
      RETURN
C
   90 FORMAT (1H0,20H ERROR IN CSPLIN     )
      END
```

```
C
      SUBROUTINE TRIB (A,B,C,X,F,NL,NU)
      DIMENSION A(2), B(2), C(2), X(2), F(2)
      X(NL)=C(NL)/B(NL)
      F(NL)=F(NL)/B(NL)
      NLP1=NL+1
      DO 10 J=NLP1,NU
      Z=1./(B(J)-A(J)*X(J-1))
      X(J)=C(J)*Z
   10 F(J)=(F(J)-A(J)*F(J-1))*Z
      NUPNL=NU+NL
      DO 20 J1=NLP1,NU
      J=NUPNL-J1
   20 F(J)=F(J)-X(J)*F(J+1)
      RETURN
      END
C
      SUBROUTINE TRIP (A,B,C,F,Q,S,J1,J2)
      DIMENSION A(3), B(3), C(3), F(3), Q(3), S(3)
      JA=J1+1
      FN=F(J2)
C   FORWARD ELIMINATION SWEEP
      Q(J1)=-C(J1)/B(J1)
      F(J1)=F(J1)/B(J1)
      S(J1)=-A(J1)/B(J1)
      DO 10 J=JA,J2
      P=1./(B(J)+A(J)*Q(J-1))
      Q(J)=-C(J)*P
      F(J)=(F(J)-A(J)*F(J-1))*P
      S(J)=-A(J)*S(J-1)*P
   10 CONTINUE
C   BACKWARD PASS
      JJ=J1+J2
      Q(J2)=0.
      S(J2)=1.
      DO 20 I=JA,J2
      J=JJ-I
      S(J)=S(J)+Q(J)*S(J+1)
   20 Q(J)=F(J)+Q(J)*Q(J+1)
      F(J2)=(FN-C(J2)*Q(J1)-A(J2)*Q(J2-1))/(C(J2)*S(J1)+A(J2)*S(J2-1)
     1  +B(J2))
C     BACKWARD ELIMINATION PASS
      DO 30 I=JA,J2
      J=JJ-I
   30 F(J)=F(J2)*S(J)+Q(J)
      RETURN
      END
      SUBROUTINE SCALC
      COMMON /CALC/ XO, XATF, BS, ICT, FLAG, PDYDX, FXF, RFN
      COMMON /INPS/ X1, X2, X3, X4, RAD, DYDX, CHORD, FUSE, AS, RADS
C
C THIS SUBRTN CALCULATES VALUES NEEDED IN THE
C EQUATIONS SOLVING FOR Y VALUES IN SBRTN BODAN
C
      READ (5,10) X1,X2,X3,X4,RAD,THETA,CHORD
      READ (5,10) RADS,FUSE,AS
      THETA=THETA*.0174533
      DYDX=TAN(THETA)
      CHORD=ABS(CHORD)
      FUSE=FUSE/2
C
C TO FIND THE Y VALUE (BS) OF CIRCLE USED IN
C THE SECANT OGIVE CALCULATIONS
C
      XBAR=AS-X2
      YS=RADS**2-XBAR**2
      YSS=SQRT(YS)
      BS=RAD-YSS
C
C TO FIND THE X VALUE (XATF) AT THE FUSE
C
      YR=ABS(BS)+FUSE
      YRR=RADS**2-YR**2
      XSS=SQRT(YRR)
      XATF=AS-XSS
C
C TO FIND THE SLOPE
C
      PDYDX=(XATF-AS)/(FUSE-BS)
      PDYDX=-PDYDX
C
C TO FIND THE X VALUE (XO) AT THE NOSECAP
C
      XSQ=SQRT(1.+PDYDX)
      RFN=FUSE*XSQ
      FXF2=RFN**2-FUSE**2
      FXF=SQRT(FXF2)
      XS=RFN-FXF
      XO=XATF-XS
      WRITE (6,20) X1,X2,X3,X4,RAD,THETA,CHORD
      WRITE (6,30) RADS,FUSE,AS,BS,XO,XATF
      RETURN
C
 10   FORMAT (8F10.0)
 20   FORMAT (1H0,27HX1,X2,X3,X4,RAD,THETA,CHORD,/,8F14.5)
 30   FORMAT (1H0,23HRADS,FUSE,AS,BS,XO,XATF,/6F14.5)
      END
      SUBROUTINE SECANT
      COMMON /CALC/ X0, XATF, BS, ICT, FLAG, PDYDX, FXF, RFN
      COMMON /INPS/ X1, X2, X3, X4, RAD, DYDX, CHORD, FUSE, AS, RADS
      COMMON JMAX, KMAX, JM, KM, N800, J800
      COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1   , T(100), TS(100)
C
C    PROJECTILE WITH NOSECAP,SECANT OGIVE,CYLINDER,BOATTAIL
C
      RCH=1./CHORD
      X1=X1*RCH
      X2=X2*RCH
      X3=X3*RCH
      X4=X4*RCH
      RAD=RAD*RCH
      FUSE=FUSE*RCH
      RADS=RADS*RCH
      XATF=XATF*RCH
      AS=AS*RCH
      BS=BS*RCH
      DO 40 J=1,J800
      IF (XX(J).GE.XATF) GO TO 10
C
C   COMPUTE Y VALUES FOR NOSECAP
C
      XBAR=XX(J)-(XATF+FXF)
      RADI=RFN**2-XBAR**2
      YY(J)=SQRT(RADI)
      GO TO 40
   10 IF (XX(J).GE.X2) GO TO 20
C
C   COMPUTE Y VALUES FOR OGIVE
C
      XBAR=XX(J)-AS
      RADIS=RADS**2-XBAR**2
      YY(J)=BS+SQRT(RADIS)
      GO TO 40
   20 IF (XX(J).GE.X3) GO TO 30
C
C   COMPUTE Y VALUES FOR CYLINDER
C
      YY(J)=RAD
      GO TO 40
C
C   COMPUTE Y VALUES FOR BOATTAIL
C
   30 YY(J)=RAD+(XX(J)-X3)*DYDX
   40 CONTINUE
      WRITE (6,50) PDYDX,RFN,FXF
      RETURN
C
   50 FORMAT (1H0,13HPDYDX,RFN,FXF,5X,3F14.5)
      END
      SUBROUTINE STING (NCGRD)
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /BODY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1   , T(100), TS(100)
C
C    STING ASSUMED ALIGNED WITH X COORDINATE
C    STING INPUT AS A SEPARATE BODY, J=1,NCGRD...BODIS DATA STARTS AT 1
      READ (5,20) ISEGS
      WRITE (6,30) ISEGS
      READ (5,40) XMIN,YMIN,XMAX,YMAX
      WRITE (6,50) XMIN,YMIN,XMAX,YMAX
      CALL BODIS (ISEGS,NCGRD)
      DO 10 JJ=2,NCGRD
      J=JBOD+JJ-1
      S(JJ)=(S(JJ)-S(1))/(S(NCGRD)-S(1))
      XX(J)=XMIN+S(JJ)*(XMAX-XMIN)
   10 YY(J)=YMIN+S(JJ)*(YMAX-YMIN)
      JBOD=JBOD+NCGRD-1
      WRITE (6,60) (XX(J),J=1,JBOD)
      RETURN
C
   20 FORMAT (I5)
   30 FORMAT (1H0,35H STING PART OF PROGRAM......ISEGS =,I5)
   40 FORMAT (8F10.0)
   50 FORMAT (1H0,21H XMIN,YMIN,XMAX,YMAX ,4F13.5)
   60 FORMAT (1H ,8F12.5)
      END
C
      SUBROUTINE CARTB (NCART)
      COMMON JMAX, KMAX, JM, KM, NBOD, JBOD
      COMMON /BODY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
     1   , T(100), TS(100)
C
      READ (5,30) ISEGS
      WRITE (6,40) ISEGS
      READ (5,50) XMIN,YMIN,XMAX,YMAX
      WRITE (6,60) XMIN,YMIN,XMAX,YMAX
C    SHIFT POINTS
      DO 10 JS=1,JBOD
      J=JBOD+NCART-JS
      JJ=JBOD+1-JS
      XX(J)=XX(JJ)
   10 YY(J)=YY(JJ)
      JBOD=JBOD+NCART-1
      CALL BODIS (ISEGS,NCART)
      NCM=NCART-1
      DO 20 J=1,NCM
      S(J)=(S(J)-S(1))/(S(NCART)-S(1))
      XX(J)=XMIN+S(J)*(XMAX-XMIN)
   20 YY(J)=YMIN+S(J)*(YMAX-YMIN)
      RETURN
C
   30 FORMAT (I5)
   40 FORMAT (1H0,33H CARTB PART OF PROGRAM.....ISEGS=,I5)
   50 FORMAT (8F10.0)
   60 FORMAT (1H0,21H XMIN,YMIN,XMAX,YMAX ,4F13.5)
      END
```

APPENDIX B

DEFINITION OF INPUT VALUES

Because the computer code has so many options, input statements are scattered throughout the code. Listed below are the input parameters to the program along with an explanation of each parameter. An input card is indicated below by numbering and underlining. All input formats are either I5 or F10.0 and are so indicated on the right hand side of the list of input parameters. Variables names follow conventional FORTRAN conventions and all integer names begin with I, J, K, L, N, or M. Special instructions as to whether or not a card is read are indicated with $ symbols.

INNER BOUNDARY

1. I3D, ND, LMAX *3I5*

I3D = 1 generates 3-dimensional data, otherwise, 2D data only is generated.

ND = same as LMAX.

LMAX = the number of planes in the circumferential direction.

2. NBOD, JBOD, IXORS, ISEGS *4I5*

NBOD = number of ordinates (i.e., x,y data points) used to define body. If NBOD < 0, an analytic body shape is used in subroutine BODAN.

JBOD = number of points user will distribute on body surface.

IXORS... use x or s (arc length) as monotone clustering parameter, x used if and only if (iff) IXORS 0.

ISEGS = number of contiguous clustering segments along body surface. Each segment requires end points and spacing specification as read in below.

$ Read cards 3, 4, 5 iff NBOD .GE.0$

3. CHORD *F10.0*

All x,y data is normalized (i.e., divided by) CHORD. CHORD may be set to 1.

4. X(J), Y(J) *2F10.0*

x,y ordinates that define body, J = 1, NBOD data cards are read in. If x,y are correctly normalized, set CHORD = 1.

5. JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*

Data that defines the cubic stretching function, see Eqs. (1) and (2) and Fig. (11).

There are ISEGS such cards read-in.

In the notation of the text

JI = j_o

JF = j_f

XI = x_o

XF = x_f

DXI = Δx_o

DXF = ∇x_f

Note DXI and DXF may both be positive or both be negative as x is increasing or decreasing

$ Read cards 6, 7, 8 iff NBOD .LT.0 $

6. TAU, FLAG *2F10.0*

TAU = parabolic arc thickness ratio
FLAG = 0 tangent-ogive cylinder, boattail projectile read in
1 secant-ogive cylinder, boattail projectile with nosecap read in

iff FLAG = 1, card 7 goes after card 8b.

7. JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*

see card 5

$ iff FLAG .GE.0$

8a. X1, X2, X3, X4, RAD, THETA, CHORD *6F10.0*

X1 = value of x at nose
X2 = value of x at ogive-cylinder juncture
X3 = value of x at cylinder-boattail juncture
X4 = value of x at boattail base
RAD = radius of cylinder
THETA = angle of degrees that boattail makes with cylinder (THETA is negative)
CHORD = if CHORD .GE.0, body length normalized to one.

Note: a spherical cap is added to boattail so the body length is not X4-X1.

8b. RADS, FUSE, AS *3F10.0*

$ iff FLAG .EQ.1$

RADS = radius of secant
FUSE = fuse height (at nosecap)
AS = value of x at secant-origin

9. NFLAG *I5*

Option to exit program after body clustering data is printed out.

NFLAG .LT.0, STOP

10. NCGRD, NCART *2I5*

Parameters that allow addition of sting/rear cut and a front cut.

NCGRD .GT.0, NCGRD points added for rear cut or sting

NCART .GT.0, NCART points added for front cut (or lower cut of C-grid)

$ Read cards 11, 12, 13 iff NCGRD .GT.0$

11. ISEGS *I5*

ISEGS of sting

12. XMIN, YMIN, XMAX, YMAX *4F10.0*

XMIN = initial x value of sting

YMIN = initial y value of sting

XMAX = final x value of sting

YMAX = YMIN

13. JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*

see card 5

$Read cards 14, 15, 16 iff NCART .GT.0$

14. ISEGS *I5*

15. XMIN, YMIN, XMAX, YMAX *4F10.0*

16. JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*

Front cut data like sting data

17. NSEGS, IOUTD *2I5*

NSEGS = number of contiguous cubic segments which are used to form an outer boundary

IOUTD = number of clustering segments along outer boundary (i.e., previous ISEGS)

18. X0, Y0, X1, Y1, TH0, TH1 *6F10.0*

x,y,θ end point values used to define cubic segment according to Eq. (4). There are NSEGS such cards read-in. Here 0 implies initial point, 1 implies final end point. The angle θ is in degrees, and is defined in the usual way. See discussion of Fig. 12 for examples of θ.

$ Read cards 19 iff IOUTD .GT.0$.

19. JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*

See card 5, there are IOUTD such data cards read in.

Arc length clustering used and, as the total arc length is not known on the first run of the program it is output. Use of normalized arc length allows user to cluster without true value of arc length.

GRID GENERATION

20. KMAX, ITERM, IPER, NCLUS, ISTOR, JELLI *6I5*

KMAX = number of points in η-direction

ITERM = number of iterations used to relax Eq. (9)
If ITERM .LT.0, straight ray grid is generated

IPER.... set IPER .GT.0 if periodic grid generated

NCLUS... NCLUS .LT.0 means grid is not reclustered using Eq. (8)

ISTOR... store grid on computer disc storage if ISTOR .GT.0

JELLI... If JELLI .GE.1, Limits JI and JF are set on the elliptic grid domain.

21. DS, OMEGA *2F10.0*

DS = Δs_0 (i.e., Δs in η direction at η = 0 boundary). See Eq. (8). Note Δs_0 used along entire η = 0 boundary.

OMEGA = relaxation factor for SLOR in Subroutine RELAX. Typical safe value is 1.55. 0 < OMEGA < 2.0.

22. JI, JF *2I5*

JELLI such cards read in. Limits of ξ min to ξ max over which an elliptic solver is used. JI and JF must correspond to vertical or horizontal rays.

23. BLANK CARD

24. BLANK CARD

$\overline{P}$ and $\overline{Q}$ input data, not recommended

APPENDIX C

SAMPLE INPUT AND OUTPUT

The following computer output illustrates the output from a sample grid generation. The tubular projectile illustrated in Figure 15 (the outer boundary is identical to that shown in Figure 13a) was used as a sample case. This particular case is the most difficult to set up as it requires the largest number of special instructions. Input values are printed after they are read in, so the output also supplies the user with an example of the data input cards.

```
I30,NO,LMAX      1     10    10

********** INNER BOUNDARY **********

NBOD,JBOD,IXORS,ISFGS     57    40     1     4

 X,Y INPUT DEFINING BODY... SUB. BODY

 NORMALIZING CHORD LENGTH INPUT          2.25000
 J,X,Y, ON BODY         1         2.25000      0.00000
 J,X,Y, ON BODY         2         2.24823      -.00628
 J,X,Y, ON BODY         3         2.23857      -.00789
 J,X,Y, ON BODY         4         2.17415      -.01674
 J,X,Y, ON BODY         5         2.09362      -.02995
 J,X,Y, ON BODY         6         1.93257      -.05652
 J,X,Y, ON BODY         7         1.77153      -.08261
 J,X,Y, ON BODY         8         1.61048      -.10838
 J,X,Y, ON BODY         9         1.44943      -.13302
 J,X,Y, ON BODY        10         1.28838      -.15621
 J,X,Y, ON BODY        11         1.12734      -.17650
 J,X,Y, ON BODY        12          .96628      -.18842
 J,X,Y, ON BODY        13          .88576      -.19116
 J,X,Y, ON BODY        14          .86965      -.19164
 J,X,Y, ON BODY        15          .85355      -.19148
 J,X,Y, ON BODY        16          .80523      -.19116
 J,X,Y, ON BODY        17          .79879      -.19068
 J,X,Y, ON BODY        18          .72471      -.18890
 J,X,Y, ON BODY        19          .64419      -.18601
 J,X,Y, ON BODY        20          .56366      -.18053
 J,X,Y, ON BODY        21          .45093      -.16684
 J,X,Y, ON BODY        22          .32209      -.14123
 J,X,Y, ON BODY        23          .24157      -.12110
 J,X,Y, ON BODY        24          .16104      -.09598
 J,X,Y, ON BODY        25          .12078      -.07553
 J,X,Y, ON BODY        26          .06441      -.02963
 J,X,Y, ON BODY        27          .02415       .02866
 J,X,Y, ON BODY        28          .00161       .07488
 J,X,Y, ON BODY        29         0.00000       .10017
 J,X,Y, ON BODY        30          .00161       .12545
 J,X,Y, ON BODY        31          .02415       .18472
 J,X,Y, ON BODY        32          .06441       .23464
 J,X,Y, ON BODY        33          .12078       .28054
 J,X,Y, ON BODY        34          .16104       .30647
 J,X,Y, ON BODY        35          .24157       .34609
 J,X,Y, ON BODY        36          .32209       .37540
 J,X,Y, ON BODY        37          .45093       .40970
 J,X,Y, ON BODY        38          .56366       .42870
 J,X,Y, ON BODY        39          .64419       .43450
 J,X,Y, ON BODY        40          .72471       .43457
 J,X,Y, ON BODY        41          .79879       .43515
 J,X,Y, ON BODY        42          .80523       .43321
 J,X,Y, ON BODY        43          .85355       .42999
 J,X,Y, ON BODY        44          .86965       .42935
 J,X,Y, ON BODY        45          .88576       .42806
```

```
J.X.Y. ON BODY     46         .96628         .42114
J.X.Y. ON BODY     47        1.12734         .40084
J.X.Y. ON BODY     48        1.28838         .36686
J.X.Y. ON BODY     49        1.44943         .32290
J.X.Y. ON BODY     50        1.61048         .26911
J.X.Y. ON BODY     51        1.77153         .20904
J.X.Y. ON BODY     52        1.93257         .14168
J.X.Y. ON BODY     53        2.09362         .07456
J.X.Y. ON BODY     54        2.17145         .04074
J.X.Y. ON BODY     57        2.25000        0.00000

JI.JF.XI.XF.DXI.DXF      1   13     1.00000      .40000     -.02000     -.06000

JI.JF.XI.XF.DXI.DXF     13   21      .40000     0.00000     -.06000     -.01500

JI.JF.XI.XF.DXI.DXF     21   28     0.00000      .40000      .01500      .06000

JI.JF.XI.XF.DXI.DXF     28   40      .40000     1.00000      .06000      .02000
    1.00000      .98000      .95091      .91382      .86982      .82000      .76545      .70727      .64655      .58436
     .52182      .46000      .40000      .34000      .27571      .21071      .14857      .09286      .04714      .01500
    0.00000      .01500      .05714      .11857      .19143      .26786      .34000      .40000      .46000      .52182
     .58436      .64655      .70727      .76545      .82000      .86982      .91382      .95091      .98000     1.00000


J AND BODY DISTRIBUTED X AND Y
J.X.Y. ON BODY      1        1.00000        0.00000
J.X.Y. ON BODY      2         .97991        -.00454
J.X.Y. ON BODY      3         .95017        -.01039
J.X.Y. ON BODY      4         .91023        -.01646
J.X.Y. ON BODY      5         .86370        -.02431
J.X.Y. ON BODY      6         .81084        -.03296
J.X.Y. ON BODY      7         .75284        -.04226
J.X.Y. ON BODY      8         .69101        -.05203
J.X.Y. ON BODY      9         .62637        -.06175
J.X.Y. ON BODY     10         .56012        -.07115
J.X.Y. ON BODY     11         .49330        -.07923
J.X.Y. ON BODY     12         .42693        -.08384
J.X.Y. ON BODY     13         .36237        -.08516
J.X.Y. ON BODY     14         .29783        -.08324
J.X.Y. ON BODY     15         .22886        -.07804
J.X.Y. ON BODY     16         .15986        -.06654
J.X.Y. ON BODY     17         .09491        -.05050
J.X.Y. ON BODY     18         .04103        -.02484
J.X.Y. ON BODY     19         .01038         .01330
J.X.Y. ON BODY     20         .00002         .04566
J.X.Y. ON BODY     21         .00188         .06164
J.X.Y. ON BODY     22         .00775         .07678
J.X.Y. ON BODY     23         .03662         .11159
J.X.Y. ON BODY     24         .09222         .14710
J.X.Y. ON BODY     25         .16595         .17365
```

```
J.X.Y. ON BODY     26        .24650        .19003
J.X.Y. ON BODY     27        .32405        .19331
J.X.Y. ON BODY     28        .38841        .19069
J.X.Y. ON BODY     29        .45272        .18482
J.X.Y. ON BODY     30        .51851        .17500
J.X.Y. ON BODY     31        .58418        .16017
J.X.Y. ON BODY     32        .64862        .14215
J.X.Y. ON BODY     33        .71061        .12143
J.X.Y. ON BODY     34        .76941        .09993
J.X.Y. ON BODY     35        .82382        .07781
J.X.Y. ON BODY     36        .87318        .05705
J.X.Y. ON BODY     37        .91688        .03895
J.X.Y. ON BODY     38        .95373        .02299
J.X.Y. ON BODY     39        .98202        .01105
J.X.Y. ON BODY     40       1.00000       0.00000

NCGRD.NCART     11   11

STING PART OF PROGRAM......ISEGS =     1

XMIN.YMIN.XMAX.YMAX        1.00000      0.00000      8.00000      0.00000

JI.JF.XI.XF.DXI.DXF      1    11      0.00000      7.00000      .02500      1.10000
   0.00000      .02500      .26111      .68542     1.27500     2.00694     2.85833     3.80625     4.82778     5.90000
   7.00000
    1.00000      .97991      .95017      .91023      .86370      .81084      .75284      .69101
     .62637      .56012      .49330      .42693      .36237      .29783      .22886      .15986
     .09491      .04103      .01038      .00002      .00188      .00775      .03662      .09222
     .16595      .24650      .32405      .38841      .45272      .51851      .58418      .64862
     .71061      .76941      .82382      .87318      .91688      .95373      .98202     1.00000
    1.02500     1.26111     1.68542     2.27500     3.00694     3.85833     4.80625     5.82778
    6.90000     8.00000

CARTB PART OF PROGRAM.....ISEGS=     1

XMIN.YMIN.XMAX.YMAX        8.00000      0.00000      1.00000      0.00000



FINAL VALUE OF JMAX     60

FINAL VALUES OF J.X.Y ALONG INNER BOUNDARY
    1      8.000000      0.000000
    2      6.900000      0.000000
    3      5.827778      0.000000
    4      4.806250      0.000000
    5      3.858333      0.000000
    6      3.006944      0.000000
    7      2.275000      0.000000
    8      1.685417      0.000000
    9      1.261111      0.000000
```

```
10      1.025000     0.000000
11      1.000000     0.000000
12       .979906     -.004536
13       .950165     -.010391
14       .910228     -.016457
15       .863697     -.024312
16       .810845     -.032953
17       .752835     -.042259
18       .691006     -.052031
19       .626374     -.061748
20       .560119     -.071150
21       .493296     -.079234
22       .426927     -.083841
23       .362371     -.085156
24       .297834     -.083235
25       .228856     -.078041
26       .159862     -.066536
27       .094908     -.050501
28       .041031     -.024843
29       .010375      .013305
30       .000018      .045659
31       .001885      .061642
32       .007749      .076784
33       .034619      .111588
34       .092225      .147097
35       .165945      .173648
36       .246503      .190027
37       .324049      .193306
38       .388408      .190688
39       .452717      .184818
40       .519509      .174999
41       .584179      .160173
42       .648621      .142148
43       .710605      .121428
44       .769414      .099933
45       .823815      .077812
46       .873175      .057050
47       .916884      .038946
48       .953733      .022992
49       .982022      .011045
50      1.000000     0.000000
51      1.025000     0.000000
52      1.261111     0.000000
53      1.685417     0.000000
54      2.275000     0.000000
55      3.006944     0.000000
56      3.858333     0.000000
57      4.806250     0.000000
58      5.827778     0.000000
59      6.900000     0.000000
```

```
********** OUTER BOUNDARY **********

JMAX.NSEGS   60    4

 X0.Y0.X1.Y1.TH0.TH1        8.00000     -1.36000     -5.00000     -1.36000    180.00000    180.00000

X0.A1.A2.Y0.B1.B2.SARC
     8.00000    -13.00000      0.00000     -1.36000      0.00000      0.00000     13.00000

 X0.Y0.X1.Y1.TH0.TH1       -5.00000     -1.36000     -7.00000      0.00000    180.00000     90.00000

X0.A1.A2.Y0.B1.B2.SARC
    -5.00000     -4.00000      2.00000     -1.36000       .00000      1.35000      2.74604

 X0.Y0.X1.Y1.TH0.TH1       -7.00000      0.00000      0.00000      8.00000     90.00000      0.00000

X0.A1.A2.Y0.B1.B2.SARC
    -7.00000       .00000      7.00000      0.00000     16.00000     -8.00000     12.18265

 X0.Y0.X1.Y1.TH0.TH1        0.00000      8.00000      8.00000      8.00000      0.00000      0.00000

X0.A1.A2.Y0.B1.B2.SARC
     0.00000      8.00000      0.00000      8.00000      0.00000      0.00000      8.00000

 OUTER BOUNDARY NONDIMENSIONAL ARC LENGTH      35.928688

 JI.JF.X1.XF.DXI.DXF      1   11      0.00000      7.00000      1.10000      .02500

 JI.JF.X1.XF.DXI.DXF     11   50      7.00000     28.92869      .02500      .02500

 JI.JF.X1.XF.DXI.DXF     50   60     28.92869     35.92869      .02500      1.10000
     0.00000    1.10000    2.17222    3.19375    4.14167    4.99306    5.72500    6.31458    6.73889    6.97500
     7.00000    7.02500    7.13483    7.32491    7.59065    7.92747    8.33078    8.79599    9.31853    9.89379
    10.51721   11.18418   11.89014   12.63049   13.40064   14.19601   15.01202   15.84408   16.68760   17.53800
    18.39069   19.24109   20.08461   20.91667   21.73268   22.52805   23.29820   24.03855   24.74451   25.41148
    26.03490   26.61016   27.13270   27.59791   28.00122   28.33804   28.60378   28.79386   28.90369   28.92869
    28.95369   29.18980   29.61411   30.20369   30.93563   31.78702   32.73494   33.75647   34.82859   35.92859
      0.00000      .03062      .06046      .08889      .11527      .13897      .15934      .17575
       .18756      .19413      .19483      .19553      .19858      .20387      .21127      .22064
       .23187      .24482      .25936      .27537      .29272      .31129      .33094      .35154
       .37298      .39512      .41783      .44099      .46446      .48813      .51187      .53554
       .55901      .58217      .60488      .62702      .64846      .66906      .68871      .70728
       .72463      .74064      .75518      .76813      .77936      .78873      .79613      .80142
       .80447      .80517      .80587      .81244      .82425      .84066      .86103      .88473
       .91111      .93954      .96938     1.00000
XS.YS          8.000000      -1.360000
XS.YS          6.900000      -1.360000
XS.YS          5.827778      -1.360000
XS.YS          4.806250      -1.360000
XS.YS          3.858334      -1.360000
XS.YS          3.006945      -1.360000
XS.YS          2.275000      -1.360000
XS.YS          1.685417      -1.360000
XS.YS      1.261112     -1.360000
XS.YS      1.025000     -1.360000
XS.YS      1.000000     -1.360000
XS.YS       .975000     -1.360000
XS.YS       .865168     -1.360000
XS.YS       .675088     -1.360000
XS.YS       .409346     -1.360000
XS.YS       .072529     -1.360000
XS.YS      -.330780     -1.360000
XS.YS      -.795003     -1.360000
XS.YS     -2.517207     -1.360000
XS.YS     -3.184184     -1.360000
XS.YS     -3.890139     -1.360000
XS.YS     -4.630485     -1.360000
XS.YS     -5.400276     -1.344820
XS.YS     -6.176874     -1.185241
XS.YS     -6.815752      -.700284
XS.YS     -6.999731       .098906
XS.YS     -6.974258       .940842
XS.YS     -6.901152      1.788348
XS.YS     -6.771242      2.630966
XS.YS     -6.574595      3.458143
XS.YS     -6.301497      4.255967
XS.YS     -5.944295      5.007065
XS.YS     -5.500658      5.691402
XS.YS     -4.976997      6.289398
XS.YS     -4.389910      6.787133
XS.YS     -3.763302      7.180745
XS.YS     -3.122659      7.476729
XS.YS     -2.490312      7.688412
XS.YS     -1.883623      7.831645
XS.YS     -1.315570      7.921821
XS.YS      -.795170      7.972613
XS.YS      -.331686      7.995400
XS.YS       .072529      8.000000
XS.YS       .409346      8.000000
XS.YS       .675088      8.000000
XS.YS       .865168      8.000000
XS.YS       .975000      8.000000
XS.YS      1.000000      8.000000
XS.YS      1.025000      8.000000
XS.YS      1.261112      8.000000
XS.YS      1.685417      8.000000
XS.YS      2.275000      8.000000
XS.YS      3.006945      8.000000
XS.YS      3.858334      8.000000
XS.YS      4.806250      8.000000
XS.YS      5.827778      8.000000
XS.YS      6.900000      8.000000
XS.YS      8.000000      8.000000
```

```
********** GRID GENERATION **********

SUR. ALGRD PRINTOUT.... KMAX.ITERM.IPER.NCLUS.ISTOR. JELLI
 35   40    0    1    1    1

DS. OMEGA         .00020       1.55000

ELLIPTIC GRID FORMED FROM  11  TO  50

EXPONENT COEFFICIENTS FOR SOURCE TERMS.PFAC.QFAC
    0.00000      0.00000      0.00000      0.00000

PC1.QC1.PC2.QC2          0.00000      0.00000      0.00000      0.00000
ITERATION NBR AND RSUM       10   .32950E+02
ITERATION NBR AND RSUM       20   .28893E+02
ITERATION NBR AND RSUM       30   .25234E+02
ITERATION NBR AND RSUM       40   .10947E+02


J  K= 2 L= 1            X          Y          Z           J  K= 2 L= 2     X          Y          Z
1                  8.00000    0.00000    0.00000          1            8.00000    0.00000    -.00020
3                  5.82778    0.00000    0.00000          3            5.82778    0.00000    -.00020
5                  3.85833    0.00000    0.00000          5            3.85833    0.00000    -.00020
7                  2.27500    0.00000    0.00000          7            2.27500    0.00000    -.00020
9                  1.26111    0.00000    0.00000          9            1.26111    0.00000    -.00020
11                 1.00000    0.00000    0.00000          11           1.00000    0.00000    -.00020
13                  .95017    0.00000    -.01039          13            .95009    0.00000    -.01058
15                  .86370    0.00000    -.02431          15            .86361    0.00000    -.02449
17                  .75284    0.00000    -.04226          17            .75274    0.00000    -.04244
19                  .62637    0.00000    -.06175          19            .62628    0.00000    -.06192
21                  .49330    0.00000    -.07923          21            .49318    0.00000    -.07940
23                  .36237    0.00000    -.08516          23            .36224    0.00000    -.08531
25                  .22886    0.00000    -.07804          25            .22870    0.00000    -.07816
27                  .09491    0.00000    -.05050          27            .09473    0.00000    -.05059
29                  .01038    0.00000     .01330          29            .01018    0.00000     .01324
31                  .00188    0.00000     .06164          31            .00170    0.00000     .06172
33                  .03662    0.00000     .11159          33            .03653    0.00000     .11177
35                  .16595    0.00000     .17365          35            .16580    0.00000     .17379
37                  .32405    0.00000     .19331          37            .32391    0.00000     .19345
39                  .45272    0.00000     .18482          39            .45261    0.00000     .18499
41                  .58418    0.00000     .16017          41            .58409    0.00000     .16035
43                  .71061    0.00000     .12143          43            .71053    0.00000     .12161
45                  .82382    0.00000     .07781          45            .82375    0.00000     .07800
47                  .91688    0.00000     .03895          47            .91683    0.00000     .03914
49                  .98202    0.00000     .01105          49            .98199    0.00000     .01124
51                 1.02500    0.00000    0.00000          51           1.02500    0.00000     .00020
53                 1.68542    0.00000    0.00000          53           1.68542    0.00000     .00020
55                 3.00694    0.00000    0.00000          55           3.00694    0.00000     .00020
57                 4.80625    0.00000    0.00000          57           4.80625    0.00000     .00020
59                 6.90000    0.00000    0.00000          59           6.90000    0.00000     .00020
```

```
  J  K= 2 L=20            X          Y          Z         J  K= 2 L=21       X              Y              Z
  1                 8.00000    0.00000    -.05087         1                 8.00000        0.00000        -.06347
  3                 5.82778    0.00000    -.05087         3                 5.82778        0.00000        -.06347
  5                 3.85833    0.00000    -.05087         5                 3.85833        0.00000        -.06347
  7                 2.27500    0.00000    -.05087         7                 2.27500        0.00000        -.06347
  9                 1.26111    0.00000    -.05087         9                 1.26111        0.00000        -.06347
 11                 1.00000    0.00000    -.05087        11                 1.00000        0.00000        -.06347
 13                  .93289    0.00000    -.05879        13                  .92932        0.00000        -.07104
 15                  .84217    0.00000    -.07148        15                  .83722        0.00000        -.08340
 17                  .72691    0.00000    -.09346        17                  .72043        0.00000        -.10675
 19                  .59315    0.00000    -.12037        19                  .58399        0.00000        -.13619
 21                  .44800    0.00000    -.14418        21                  .43441        0.00000        -.16219
 23                  .29871    0.00000    -.15127        23                  .27854        0.00000        -.16969
 25                  .14209    0.00000    -.13673        25                  .11398        0.00000        -.15278
 27                 -.01283    0.00000    -.03977        27                 -.04831        0.00000        -.09786
 29                 -.10367    0.00000     .00460        29                 -.14003        0.00000         .00802
 31                 -.10416    0.00000     .11023        31                 -.13833        0.00000         .12469
 33                 -.04096    0.00000     .20107        33                 -.07122        0.00000         .22447
 35                  .08073    0.00000     .26135        35                  .05300        0.00000         .28935
 37                  .23974    0.00000     .28444        37                  .21321        0.00000         .31460
 39                  .38626    0.00000     .29032        39                  .36369        0.00000         .32382
 41                  .52775    0.00000     .27132        41                  .50926        0.00000         .30722
 43                  .66148    0.00000     .23578        43                  .64617        0.00000         .27304
 45                  .78153    0.00000     .19451        45                  .76498        0.00000         .23264
 47                  .88350    0.00000     .15796        47                  .87411        0.00000         .19675
 49                  .96589    0.00000     .13278        49                  .96163        0.00000         .17215
 51                 1.02500    0.00000     .12258        51                 1.02500        0.00000         .16211
 53                 1.68542    0.00000     .12258        53                 1.68542        0.00000         .16211
 55                 3.00694    0.00000     .12258        55                 3.00694        0.00000         .16211

  J  K= 6 L= 1            X          Y          Z         J  K= 6 L= 2       X              Y              Z
  1                 8.00000    0.00000    0.00000         1                 8.00000        -.00019         .00004
  3                 5.82778    0.00000    0.00000         3                 5.82778        -.00019         .00004
  5                 3.85833    0.00000    0.00000         5                 3.85833        -.00019         .00004
  7                 2.27500    0.00000    0.00000         7                 2.27500        -.00019         .00004
  9                 1.26111    0.00000    0.00000         9                 1.26111        -.00019         .00004
 11                 1.00000    0.00000    0.00000        11                 1.00000        -.00019         .00004
 13                  .95017    -.01013     .00231        13                  .95009        -.01031         .00235
 15                  .86370    -.02370     .00541        15                  .86361        -.02388         .00545
 17                  .75284    -.04120     .00940        17                  .75274        -.04137         .00944
 19                  .62637    -.06070     .01374        19                  .62628        -.06037         .01378
 21                  .49330    -.07725     .01763        21                  .49318        -.07741         .01767
 23                  .36237    -.08302     .01895        23                  .36224        -.08317         .01898
 25                  .22886    -.07608     .01737        25                  .22870        -.07620         .01739
 27                  .09491    -.04923     .01124        27                  .09473        -.04932         .01126
 29                  .01038     .01297    -.00296        29                  .01018         .01291        -.00295
 31                  .00188     .06010    -.01372        31                  .00170         .06018        -.01373
 33                  .03662     .10879    -.02483        33                  .03653         .10896        -.02487
 35                  .16595     .16929    -.03864        35                  .16580         .16943        -.03867
 37                  .32405     .18846    -.04301        37                  .32391         .18860        -.04305
 39                  .45272     .18018    -.04113        39                  .45261         .18035        -.04116
 41                  .58418     .15616    -.03564        41                  .58409         .15633        -.03568
 43                .71061     .11838     -.02702      43                 .71053     .11856     -.02706
 45                .82382     .07586     -.01731      45                 .82375     .07504     -.01736
 47                .91688     .03797     -.00867      47                 .91683     .03816     -.00871
 49                .98202     .01077     -.00246      49                 .98199     .01096     -.00250
 51               1.02500    0.00000     0.00000      51                1.02500     .00019     -.00004
 53               1.68542    0.00000     0.00000      53                1.68542     .00019     -.00004
 55               3.00694    0.00000     0.00000      55                3.00694     .00019     -.00004
 57               4.80625    0.00000     0.00000      57                4.80625     .00019     -.00004
 59               6.90000    0.00000     0.00000      59                6.90000     .00019     -.00004

  J  K= 6 L=20      X          Y          Z          J  K= 6 L=21     X          Y          Z
  1               8.00000    -.04960     .01132       1                8.00000    -.06187     .01412
  3               5.82778    -.04960     .01132       3                5.82778    -.06187     .01412
  5               3.85833    -.04960     .01132       5                3.85833    -.06187     .01412
  7               2.27500    -.04960     .01132       7                2.27500    -.06187     .01412
  9               1.26111    -.04960     .01132       9                1.26111    -.06187     .01412
 11               1.00000    -.04960     .01132      11                1.00000    -.06187     .01412
 13                .93289    -.05731     .01308      13                 .92932    -.06926     .01581
 15                .84217    -.06968     .01590      15                 .83722    -.08131     .01856
 17                .72691    -.09111     .02080      17                 .72063    -.10408     .02375
 19                .59315    -.11736     .02679      19                 .58399    -.13278     .03031
 21                .44800    -.14057     .03208      21                 .43441    -.15813     .03609
 23                .29871    -.14747     .03366      23                 .27854    -.16543     .03776
 25                .14209    -.13330     .03043      25                 .11398    -.14895     .03400
 27               -.01283    -.08752     .01998      27                -.04831    -.09541     .02178
 29               -.10367     .00448    -.00102      29                -.14003     .00782    -.00179
 31               -.10416     .10746    -.02453      31                -.13833     .12156    -.02775
 33               -.04096     .19603    -.04474      33                -.07122     .21884    -.04995
 35                .08073     .25480    -.05816      35                 .05300     .28210    -.06439
 37                .23974     .27731    -.06329      37                 .21321     .30672    -.07001
 39                .38626     .28304    -.06460      39                 .36369     .31570    -.07206
 41                .52775     .26452    -.06038      41                 .50926     .29951    -.06836
 43                .66148     .22987    -.05247      43                 .64617     .26620    -.06076
 45                .78153     .18963    -.04328      45                 .76898     .22681    -.05177
 47                .88350     .15400    -.03515      47                 .87411     .19182    -.04378
 49                .96589     .12946    -.02955      49                 .96163     .16783    -.03831
 51               1.02500     .11951    -.02728      51                1.02500     .15804    -.03607
 53               1.68542     .11951    -.02728      53                1.68542     .15804    -.03607
 55               3.00694     .11951    -.02728      55                3.00694     .15804    -.03607
 57               4.80625     .11951    -.02728      57                4.80625     .15804    -.03607
 59               6.90000     .11951    -.02728      59                6.90000     .15804    -.03607

  J  K=10 L= 1      X          Y          Z          J  K=10 L= 2     X          Y          Z
  1               8.00000    0.00000    0.00000       1                8.00000     .00009     .00018
  3               5.82778    0.00000    0.00000       3                5.82778     .00009     .00018
  5               3.85833    0.00000    0.00000       5                3.85833     .00009     .00018
  7               2.27500    0.00000    0.00000       7                2.27500     .00009     .00018
  9               1.26111    0.00000    0.00000       9                1.26111     .00009     .00018
 11               1.00000    0.00000    0.00000      11                1.00000     .00009     .00018
 13                .95017     .00451     .00935      13                 .95009     .00459     .00953
 15                .86370     .01055     .02190      15                 .86361     .01063     .02207
 17                .75284     .01834     .03807      17                 .75274     .01841     .03823
 19                .62637     .02679     .05563      19                 .62628     .02687     .05579
```

```
21                .49330     .03438     .07139
23                .36237     .03695     .07672
25                .22886     .03386     .07031
27                .09491     .02191     .04550
29                .01038    -.00577    -.01199
31                .00188    -.02675    -.05554
33                .03662    -.04842    -.10054
35                .16595    -.07534    -.15645
37                .32405    -.08387    -.17416
39                .45272    -.08019    -.16651
41                .58418    -.06950    -.14431
43                .71061    -.05269    -.10940
45                .82382    -.03376    -.07011
47                .91688    -.01690    -.03509
49                .98202    -.00479    -.00995
51               1.02500    0.00000    0.00000
53               1.68542    0.00000    0.00000
55               3.00694    0.00000    0.00000
57               4.80625    0.00000    0.00000
59               6.90000    0.00000    0.00000

 J  K=10 L=20       X          Y          Z
 1               8.00000     .02207     .04583
 3               5.82778     .02207     .04583
 5               3.85833     .02207     .04583
 7               2.27500     .02207     .04583
 9               1.26111     .02207     .04583
11               1.00000     .02207     .04583
13                .93289     .02551     .05297
15                .84217     .03101     .06440
17                .72691     .04055     .08420
19                .59315     .05223     .10845
21                .44800     .06256     .12991
23                .29871     .06563     .13629
25                .14209     .05933     .12319
27               -.01283     .03895     .08088
29               -.10367    -.00199    -.00414
31               -.10416    -.04783    -.09931
33               -.04096    -.08724    -.18116
35                .08073    -.11340    -.23547
37                .23974    -.12341    -.25627
39                .38626    -.12597    -.26157
41                .52775    -.11772    -.24445
43                .66148    -.10230    -.21243
45                .78153    -.08439    -.17524
47                .88350    -.06854    -.14231
49                .96589    -.05761    -.11963
51               1.02500    -.05319    -.11044
53               1.68542    -.05319    -.11044
55               3.00694    -.05319    -.11044
57               4.80625    -.05319    -.11044
59               6.90000    -.05319    -.11044
```

```
21                .49318     .03445     .07154
23                .36224     .03701     .07686
25                .22870     .03391     .07042
27                .09473     .02195     .04558
29                .01018    -.00575    -.01193
31                .00170    -.02678    -.05561
33                .03653    -.04849    -.10070
35                .16580    -.07540    -.15658
37                .32391    -.08394    -.17429
39                .45261    -.08026    -.16667
41                .58409    -.06957    -.14447
43                .71053    -.05277    -.10957
45                .82375    -.03384    -.07027
47                .91683    -.01698    -.03526
49                .98199    -.00488    -.01013
51               1.02500    -.00009    -.00018
53               1.68542    -.00009    -.00018
55               3.00694    -.00009    -.00018
57               4.80625    -.00009    -.00018
59               6.90000    -.00009    -.00018

 J  K=10 L=21    X          Y          Z
 1               8.00000     .02754     .05718
 3               5.82778     .02754     .05718
 5               3.85833     .02754     .05718
 7               2.27500     .02754     .05718
 9               1.26111     .02754     .05718
11               1.00000     .02754     .05718
13                .92932     .03082     .06400
15                .83722     .03619     .07514
17                .72043     .04632     .09618
19                .58399     .05909     .12271
21                .43441     .07037     .14613
23                .27854     .07362     .15288
25                .11398     .06529     .13765
27               -.04831     .04246     .08817
29               -.14003    -.00348    -.00723
31               -.13833    -.05410    -.11234
33               -.07122    -.09739    -.20224
35                .05300    -.12555    -.26070
37                .21321    -.13650    -.28345
39                .36369    -.14050    -.29175
41                .50926    -.13330    -.27679
43                .64617    -.11847    -.24600
45                .76898    -.10094    -.20960
47                .87411    -.08537    -.17727
49                .96163    -.07469    -.15510
51               1.02500    -.07034    -.14605
53               1.68542    -.07034    -.14605
55               3.00694    -.07034    -.14605
57               4.80625    -.07034    -.14605
59               6.90000    -.07034    -.14605
```

APPENDIX D

PLOT PROGRAM LISTING

A listing of the computer code used to generate grid plots is presented in this appendix. The plot program is written in standard FORTRAN IV and uses the Tektronix Plot 10 software package. All plots were produced on the Tektronix 4010-1 display terminal which was connected to the BRL Cyber 173/76. The program is an interactive plotting routine which prompts the user for all requested information.

```
      PROGRAM IGP (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,PL3,KBI
      COMMON /GRIDC/ GXMIN, GXMAX, GYMIN, GYMAX, X(100,50), Y(100,50)
      COMMON /HOR/ TITLE(5)
C
C*****GRIP-PROGRAM TO PLOT THE COMPUTATIONAL GRID ABOUT AN
C     AERODYNAMIC BODY OR AIRFOIL SECTION******
C
C*****INPUT SECTION*****
C*****CONTROL PARAMETER INPUT
C**NADD=-1 SUPPRESSES LISTING OF PLOT DATA,NADD=1 ALLOWS IT
C**JMAX,KMAX-NUMBER OF J,K POINTS
C**HEAD-ALPHANUMERIC INFORMATION DESCRIBING THE
C     CONFIGURATION BEING PLOTTED
      CALL CONNEC (5LTAPE5)
      CALL CONNEC (5LTAPE6)
      CALL TERM (1,1024)
      CALL SETBUF (3)
      CALL ANMODE
C     READ(5,10)HEAD
C
      WRITE (6,10)
      READ (5,*) IBD
      IBD=IBD/10
      CALL INITT (IBD)
      CALL BINITT
C
      WRITE (6,20)
      READ (5,*) JMAX,KMAX
C
      WRITE (6,30)
      READ (5,*) GXMIN,GXMAX,GYMIN,GYMAX
      WRITE (6,40)
      READ (5,50) TITLE
      CALL NEWPAG
C
C***READ IN X AND Y VALUES
      READ (8) ((X(J,K),J=1,JMAX),K=1,KMAX),((Y(J,K),J=1,JMAX),K=1,KMA
C***** PLOT THE GRID *****
      CALL GRDPLT (JMAX,KMAX,IBD)
C***TERMINATE PLOTTING
      CALL FINITT (0,700)
      STOP
C
   10 FORMAT (19H WHAT IS BAUD RATE?)
   20 FORMAT (20H WHAT ARE JMAX,KMAX?)
   30 FORMAT (30H WHAT ARE XMIN,XMAX,YMIN,YMAX?)
   40 FORMAT (28H ENTER TITLE - UP TO 50 CHAR)
   50 FORMAT (5A10)
      END
      SUBROUTINE GRDPLT (JMAX,KMAX,IBD)
      COMMON /GRIDC/ GXMIN, GXMAX, GYMIN, GYMAX, X(100,50), Y(100,50)
      COMMON /HDR/ TITLE(5)
      DIMENSION GX(128), GY(128)
C
C***READJUST PLOT LIMITS IN ORDER TO AVOID STRETCHED PLOTS
      ICOUNT=0
   10 XMAX=GXMAX
      XMIN=GXMIN
      YMAX=GYMAX
      YMIN=GYMIN
      XDIF=XMAX-XMIN
      YDIF=YMAX-YMIN
      IF (XDIF.LT.YDIF) GO TO 20
      XDIFH=XDIF*0.5
      YMID=(YMAX+YMIN)*0.5
      YMX=YMID+XDIFH
      YMN=YMID-XDIFH
      YMAX=YMX
      YMIN=YMN
      GO TO 30
   20 YDIFH=YDIF*0.5
      XMID=(XMAX+XMIN)*0.5
      XMX=XMID+YDIFH
      XMN=XMID-YDIFH
      XMAX=XMX
      XMIN=XMN
   30 CONTINUE
C
C     PLOT THE LINES
      IF (ICOUNT.GT.0) GO TO 40
      AXMIN=XMIN
      AXMAX=XMAX
      AYMIN=YMIN
      AYMAX=YMAX
   40 CONTINUE
C
      CALL BINITT
      CALL NPTS (JMAX)
      CALL ANMODE
      WRITE (6,120) TITLE
      CALL XFRM (2)
      CALL YFRM (2)
C
C
      CALL DLIMX (XMIN,XMAX)
      CALL DLIMY (YMIN,YMAX)
      CALL SLIMX (150,800)
      CALL SLIMY (50,700)
C
      DO 70 K=1,KMAX
      DO 50 J=1,JMAX
      GX(J)=X(J,K)
   50 GY(J)=Y(J,K)
C
      IF (K.GT.1) GO TO 60
      CALL CHECK (GX,GY)
      CALL DSPLAY (GX,GY)
   60 CALL CPLOT (GX,GY)
   70 CONTINUE
      CALL NPTS (KMAX)
      DO 90 J=1,JMAX
      DO 80 K=1,KMAX
      GX(K)=X(J,K)
   80 GY(K)=Y(J,K)
      CALL CPLOT (GX,GY)
   90 CONTINUE
      CALL BELL
      CALL TSEND
      CALL TINPUT (II)
C
      IF (II.EQ.83) GO TO 110
C
      CALL NEWPAG
      CALL TSEND
      DO 100 J=1,JMAX
      GX(J)=X(J,1)
  100 GY(J)=Y(J,1)
      CALL BINITT
      CALL DLIMX (AXMIN,AXMAX)
      CALL DLIMY (AYMIN,AYMAX)
      CALL SLIMX (150,800)
      CALL SLIMY (50,700)
      CALL NPTS (JMAX)
      CALL XFRM (2)
      CALL YFRM (2)
      CALL CHECK (GX,GY)
      CALL DSPLAY (GX,GY)
      CALL TSEND
      CALL BELL
      CALL ANMODE
      WRITE (6,130)
      CALL TSEND
      CALL RECOVR
      CALL VCURSR (ICH,XX,YY)
      GXMIN=XX
      GYMIN=YY
      CALL ANMODE
      WRITE (6,140)
      CALL RECOVR
      CALL TSEND
      CALL ANMODE
      CALL VCURSR (ICH,XX,YY)
      GXMAX=XX
      GYMAX=YY
      CALL TSEND
      ICOUNT=ICOUNT+1
      CALL NEWPAG
      GO TO 10
  110 CONTINUE
      CALL NEWPAG
      RETURN
C
  120 FORMAT (5A10)
  130 FORMAT (30H POSITION CURSOR FOR XMIN,YMIN)
  140 FORMAT (1H0,30H POSITION CURSOR FOR XMAX,YMAX)
      END
```

DISTRIBUTION LIST

No. of Copies — Organization

12 Commander
Defense Technical Info Center
ATTN: DDC-DDA
Cameron Station
Alexandria, VA 22314

1 Commander
US Army Materiel Development
and Readiness Command
ATTN: DRCDMD-ST
5001 Eisenhower Avenue
Alexandria, VA 22333

8 Commander
US Army Armament Research and
Development Command
ATTN: DRDAR-TSS (2 cys)
DRDAR-LCA-F
Mr. D. Mertz
Mr. A. Loeb
Mr. S. Wasserman
Mr. H. Hudgins
Mr. E. Friedman
Dover, NJ 07801

1 Commander
US Army Armament Materiel
Readiness Command
ATTN: DRSAR-LEP-L, Tech Lib
Rock Island, IL 61299

1 Director
US Army Armament Research and
Development Command
ATTN: DRDAR-LCB-TL
Watervliet, NY 12189

1 Commander
US Army Aviation Research and
Development Command
ATTN: DRDAV-E
4300 Goodfellow Boulevard
St. Louis, MO 63120

No. of Copies — Organization

1 Director
US Army Air Mobility Research
and Development Laboratory
Ames Research Center
Moffett Field, CA 94035

1 Commander
US Army Communications
Research and Development
Command
ATTN: DRDCO-PPA-SA
Fort Monmouth, NJ 07703

1 Commander
US Army Electronics Research
and Development Command
Technical Support Activity
ATTN: DELSD-L
Fort Monmouth, NJ 07703

3 Commander
US Army Missile Command
ATTN: DRSMI-R
DRSMI-YDL
DRSMI-RDK
Mr. R. Deep
Redstone Arsenal, AL 35809

1 Commander
US Army Tank Automotive
Research and Development
Command
ATTN: DRDTA-UL
Warren, MI 48090

1 Director
US Army TRADOC Systems
Analysis Activity
ATTN: ATAA-SL, Tech Lib
White Sands Missile Range,
NM 88002

DISTRIBUTION LIST

No. of Copies — Organization

1 Commander
US Army Research Office
P. O. Box 12211
Research Triangle Park
NC 27709

1 Commander
US Naval Air Systems Command
ATTN: AIR-604
Washington, D. C. 20360

4 Commander
US Naval Surface Weapons Center
ATTN: Dr. T. Clare, Code DK20
Dr. P. Daniels
Mr. D. A. Jones III
Mr. L. Mason
Dahlgren, VA 22448

4 Commander
US Naval Surface Weapons Center
ATTN: Code 312
Mr. R. Voisinet
Mr. R. Driftmeyer
Mr. J. Knott
Mr. R. Schlie
Silver Spring, MD 20910

1 Commander
US Naval Weapons Center
ATTN: Code 3431, Tech Lib
China Lake, CA 93555

1 Director
NASA Langley Research Center
ATTN: NS-185, Tech Lib
Langley Station
Hampton, VA 23365

No. of Copies — Organization

3 Director
NASA Ames Research Center
ATTN: MS-202A-14
Dr. P. Kutler
MS-202-1,
Dr. T. Pulliam
MS-227-8,
Dr. L. Schiff
Moffett Field, CA 94035

2 Sandia Laboratories
ATTN: Division No. 1331,
Mr. H.R. Vaughn
Mr. G.R. Eisler
P.O. Box 580
Albuquerque, NJ 87184

1 University of California, Davis
Department of Mechanical Engineering
ATTN: Prof. H.A. Dwyer
Davis, CA 95616

1 University of Colorado
Department of Aerospace Engineering
ATTN: Prof. G. Inger
Boulder, CO 80309

1 University of Delaware
Mechanical and Aerospace Engineering Department
ATTN: Dr. J. E. Danberg
Newark, DE 19711

DISTRIBUTION LIST

No. of Copies   Organization

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, H. Cohen

Cdr, USATECOM
ATTN: DRSTE-TO-F

Dir, USACSL, Bldg. E3516, EA
ATTN: DRDAR-CLB-PA
DRDAR-ACW, Mr. M. Miller

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.)

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.)

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name:

Telephone Number:

Organization Address: